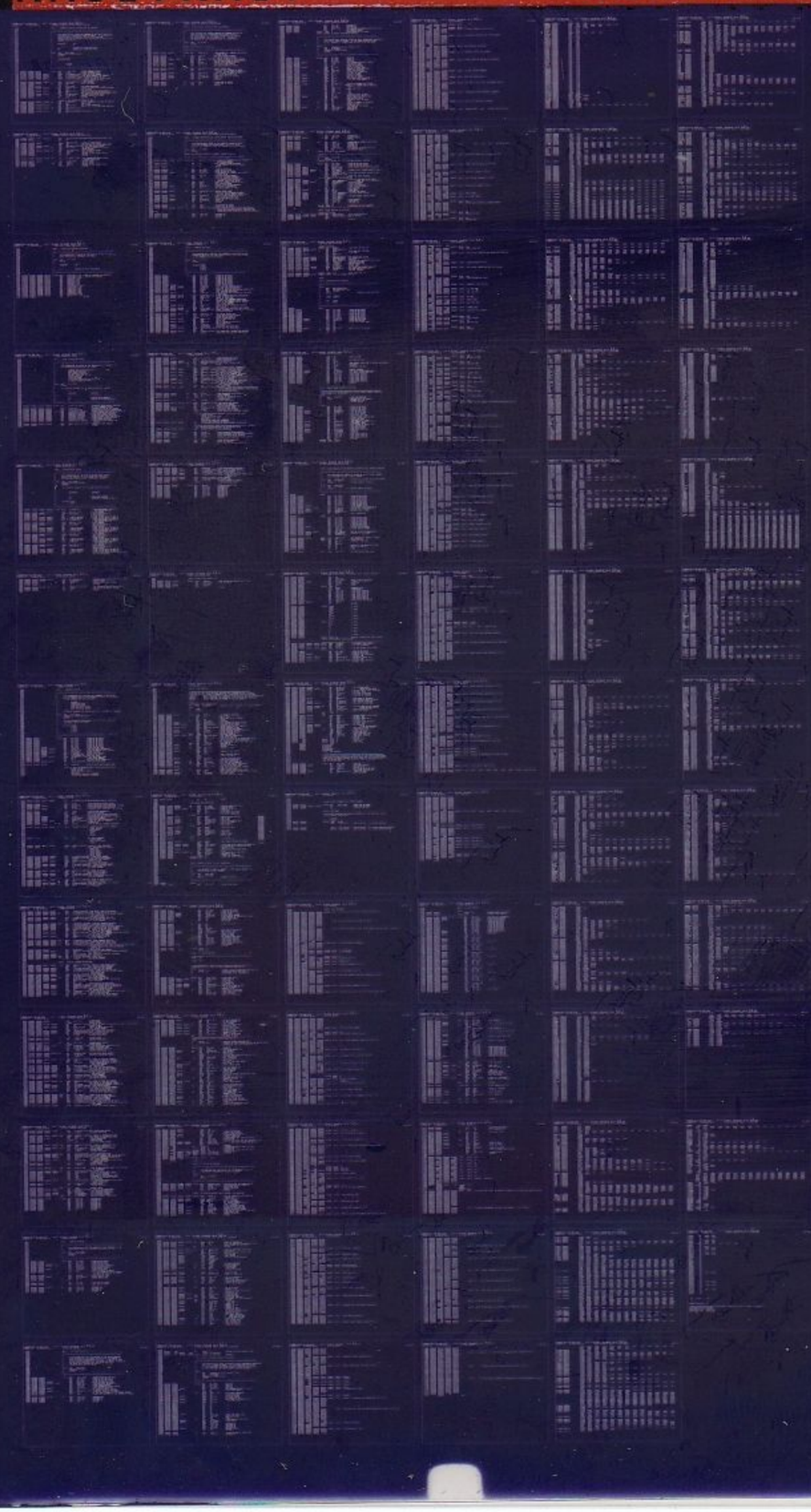
RK611
RK06,RK07

RK611/06 PERF EXER
CZR6PDO

AH-9145D-MC
FICHE 1 OF 2

RK611 RK611/06 PERF EXER AH-9145D-MC JUL 1982
RK06,RK07 CZR6PDO FICHE 2 OF 2 COPYRIGHT© 76-82
MADE IN USA



.REM @

IDENTIFICATION

PRODUCT CODE: AC-9144D-MC

PRODUCT NAME: CZR6PDO RK611/06/07 PERF EXER

DATE: JANUARY 1982

MAINTAINER: STORAGE SYSTEM SOFTWARE TEST APPLICATION

AUTHOR: B. T. LEBLANC

```
     1
```

```
43
44
45                                  REVISION HISTORY
46
47          REVISION        CHANGES                                     DATE
48
49          CZR6PD0         IMPLEMENTED XXDP LOAD MEDIA OPTION          JAN 82
50                          FIXED TO WORK UNDER ACT
```

TABLE OF CONTENTS

1.0 ABSTRACT

2.0 REQUIREMENTS
  2.1 EQUIPMENT
  2.2 PRELIMINARY PREREQUISITS

3.0 OPERATING PROCEDURES
  3.1 LOADING PROCEDURES
  3.2 STARTING PROCEDURE
  3.3 SYSTEM PARAMETERS
  3.4 SWITCHES
    3.4.1 SWITCH 15
    3.4.2 SWITCH 14
    3.4.3 SWITCH 13
    3.4.4 SWITCH 12
    3.4.5 SWITCH 11
    3.4.6 SWITCH 10
    3.4.7 SWITCH 9
    3.4.8 SWITCH 8
    3.4.9 SWITCH 7
    3.4.10 SWITCH 6
    3.4.11 SWITCH 5
    3.4.12 SWITCH 4
    3.4.13 SWITCH 3
    3.4.14 SWITCH 2
    3.4.15 SWITCH 1
    3.4.16 SWITCH 0
  3.5 RUN TIME

4.0 OPERATING PROCEDURE
  4.1 OPERATOR COMMANDS
  4.2 DRIVE PARAMETER (PER EACH DRIVE)
    4.2.1 FORMS FOR PACK AREA EXCLUSIONS
  4.3 DIAGNOSTIC DUMP

5.0 PROGRAM DESCRIPTION

6.0 PRINT OUTS
  6.1 PERFORMANCE SUMMARY TYPEOUT
  6.2 ERROR REPORTS

APPENDIX A - DATA WRITTEN ON PACKS

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 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 CF 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.

THE PROGRAM IS NOW IN AN IDLE LOOP WAITING FOR THE INITIATION OF RK06 DRIVE TESTING.

THE OPERATOR THEN STARTS PERFORMANCE EVALUATION OF THE RK06 DRIVE(S) BY TYPING A CONTROL-C <C> FOLLOWED BY A TN OR PN AS DESCRIBED IN 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 CPERATOR 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 RUNNING THIS PROGRAM FOR AN HOUR OR LESS SINCE STATISTICS ARE TAKEN AS 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 ISSUE) 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 CANCELS THE LAST CHARACTER TYPED. A CONTROL U <^U> CANCELS LINE BEING 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 COMMANC 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 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.

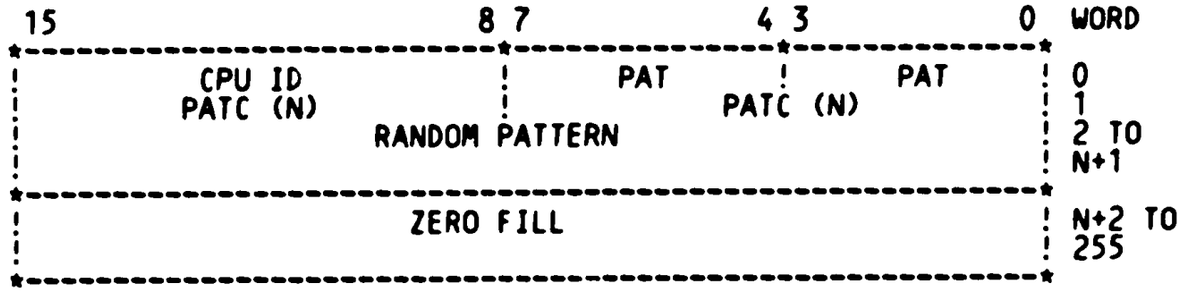
6.2 ERROR REPORTS

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

827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866

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	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

	PATTERN 4	PATTERN 5	PATTERN 6	PATTERN 7
867				
868				
869				
870				
871				
872	000000	000000	000001	000001
873	010421	177777	000003	000002
874	021042	000000	000007	000004
875	031463	000000	000017	000010
876	042104	177777	000037	000020
877	052525	177777	000077	000040
878	063146	000000	000177	000100
879	073567	000000	000377	000200
880	104210	000000	000777	000400
881	114631	177777	001777	001000
882	125252	177777	003777	002000
883	135673	177777	007777	004000
884	146314	000000	017777	010000
885	156735	000000	037777	020000
886	167356	000000	077777	040000
887	177777	000000	177777	100000
888	177777	177777	177776	100000
889	167356	177777	177774	040000
890	156735	177777	177770	020000
891	146314	177777	177760	010000
892	135673	000000	177740	004000
893	125252	000000	177700	002000
894	114631	000000	177600	001000
895	104210	177777	177400	000400
896	073567	177777	177000	000200
897	063146	177777	176000	001000
898	052525	000000	174000	000040
899	042104	000000	170000	000020
900	031463	177777	160000	000010
901	021042	177777	140000	000004
902	010421	000000	100000	000002
903	000000	177777	000000	000001

	PATTERN 10	PATTERN 11	PATTERN 12	PATTERN 13
904				
905				
906				
907				
908				
909	177776	172666	153333	000000
910	177775	155555	066667	177777
911	177773	172666	153333	177777
912	177767	155555	066667	177777
913	177757	172666	153333	177777
914	177737	155555	066667	177777
915	177677	172666	153333	177777
916	177577	155555	066667	177777
917	177377	172666	153333	177777
918	176777	155555	066667	177777
919	175777	172666	153333	177777
920	173777	155555	066667	177777
921	167777	172666	153333	177777
922	157777	155555	066667	177777
923	137777	172666	153333	177777
924	077777	155555	066667	177777
925	077777	172666	153333	177777
926	137777	155555	066667	177777
927	157777	172666	153333	177777
928	167777	155555	066667	177777
929	173777	172666	153333	177777
930	175777	155555	066667	177777
931	176777	172666	153333	177777
932	177377	155555	066667	177777
933	177577	172666	153333	177777
934	177677	155555	066667	177777
935	177737	172666	153333	177777
936	177757	155555	066667	177777
937	177767	172666	153333	177777
938	177773	155555	066667	177777
939	177775	172666	153333	177777
940	177776	155555	066667	177777

	PATTERN 14	PATTERN 15	PATTERN 16	PATTERN 17
941				
942				
943				
944				
945				
946	177777	172304	070627	133467
947	000000	172304	113431	133467
948	000000	172304	014561	133467
949	000000	172304	070627	133467
950	000000	172304	113431	133467
951	000000	172304	014561	133467
952	000000	172304	070627	133467
953	000000	172304	113431	133467
954	000000	172304	014561	133467
955	000000	172304	070627	133467
956	000000	172304	113431	133467
957	000000	172304	014561	133467
958	000000	172304	070627	133467
959	000000	172304	113431	133467
960	000000	172304	014561	133467
961	000000	172304	070627	133467
962	000000	172304	113431	133467
963	000000	172304	014561	133467
964	000000	172304	070627	133467
965	000000	172304	113431	133467
966	000000	172304	014561	133467
967	000000	172304	070627	133467
968	000000	172304	113431	133467
969	000000	172304	014561	133467
970	000000	172304	070627	133467
971	000000	172304	113431	133467
972	000000	172304	014561	133467
973	000000	172304	070627	133467
974	000000	172304	113431	133467
975	000000	172304	014561	133467
976	000000	172304	070627	133467
977	000000	172304	113431	133467
978	@			

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

                        .TITLE  CZR6PD0 RK611/06 PERF EXEC
                        ;*COPYRIGHT (C) 1976,1982
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ROY SPITZER
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C) FEB 1982.
                        ;*
                        ;;**********************************************************************
                        ;*
                        ;*      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
                        .FQUIV  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
```

```
1079            000002                  SW01=   2
1080            000001                  SW00=   1
1081                                    .EQUIV  SW09,SW9
1082                                    .EQUIV  SW08,SW8
1083                                    .EQUIV  SW07,SW7
1084                                    .EQUIV  SW06,SW6
1085                                    .EQUIV  SW05,SW5
1086                                    .EQUIV  SW04,SW4
1087                                    .EQUIV  SW03,SW3
1088                                    .EQUIV  SW02,SW2
1089                                    .EQUIV  SW01,SW1
1090                                    .EQUIV  SW00,SW0
1091
1092                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1093            100000                  BIT15=  100000
1094            040000                  BIT14=  40000
1095            020000                  BIT13=  20000
1096            010000                  BIT12=  10000
1097            004000                  BIT11=  4000
1098            002000                  BIT10=  2000
1099            001000                  BIT09=  1000
1100            000400                  BIT08=  400
1101            000200                  BIT07=  200
1102            000100                  BIT06=  100
1103            000040                  BIT05=  40
1104            000020                  BIT04=  20
1105            000010                  BIT03=  10
1106            000004                  BIT02=  4
1107            000002                  BIT01=  2
1108            000001                  BIT00=  1
1109                                    .EQUIV  BIT09,BIT9
1110                                    .EQUIV  BIT08,BIT8
1111                                    .EQUIV  BIT07,BIT7
1112                                    .EQUIV  BIT06,BIT6
1113                                    .EQUIV  BIT05,BIT5
1114                                    .EQUIV  BIT04,BIT4
1115                                    .EQUIV  BIT03,BIT3
1116                                    .EQUIV  BIT02,BIT2
1117                                    .EQUIV  BIT01,BIT1
1118                                    .EQUIV  BIT00,BIT0
1119
1120                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1121            000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1122            000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1123            000014                  TBITVEC=14              ;;"T" BIT
1124            000014                  TRTVEC= 14              ;;TRACE TRAP
1125            000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1126            000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1127            000024                  PWRVEC= 24              ;;POWER FAIL
1128            000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1129            000034                  TRAPVEC=34              ;;"TRAP" TRAP
1130            000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
1131            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1132            000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1133            000240                  APRIOR= PR5                     ;RK611 PRIOIRITY
1134            120210                  AVECT1= 120210                  ;RK611 VECTOR
```

```
                        ABASE=  177440          ;RK611 BASE
                        MEMVEC= 114             ;VECTOR FOR MEMORY CHECK ENABLE
                        MEMBAS= 172100          ;BUS ADDRESS FOR MEMORY CHECK ENABLE
                        PAR.EN= 1               ;MEMORY ENABLE MEMORY CHECKING

                        .SBTTL  RK06-RK07 CONTROLLER REGISTER DEFINITION

                        RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
                        RKWC=   2               ;WORD COUNT REGISTER
                        RKBA=   4               ;BUS ADDRESS REGISTER
                        RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
                        RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
                        RKDS=   12              ;DRIVE STATUS REGISTER
                        RKER=   14              ;ERROR REGISTER
                        RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
                        RKDC=   20              ;DESIRED CYLINDER REGISTER
                        RKDCYL= 20              ;DESIRED CYLINDER REGISTER
                        RKDB=   24              ;DATA BUFFER
                        RKMR1=  26              ;MAINTENANCE REGISTER 1
                        RKMR2=  34              ;MAINTENANCE REGISTER 2
                        RKMR3=  36              ;MAINTENANCE REGISTER 3
                        RKPOS=  30              ;ECC POSITION INFORMATION
                        RKECPS= 30              ;ECC POSITION INFORMATION
                        RKPAT=  32              ;ECC PATTERN INFORMATION
                        RKECPT= 32              ;ECC PATTERN INFORMATION

                        .SBTTL  DRIVE COMMANDS

                        SELDRV= 101             ;SELECT DRIVE
                        PACK=   103             ;PACK ACKNOWLEDGE
                        CLEAR=  105             ;DRIVE CLEAR
                        UNLOAD= 107             ;UNLOAD
                        SRTSPL= 111             ;START SPINDLE
                        RECAL=  113             ;RECALIBRATE
                        OFFSET= 115             ;OFFSET
                        SEEK=   117             ;SEEK
                        RDDATA= 121             ;READ DATA
                        WRDATA= 123             ;WRITE DATA
                        RDHEAD= 125             ;READ HEADER
                        WRHEAD= 127             ;WRITE HEADER AND DATA
                        WRTCHK= 131             ;WRITE CHECK

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

                        RELEAS= 140             ;RELEASE DRIVE
                        RDSTAT= 141             ;GET ALL STATUS FROM DRIVE
                        RDALHD= 164             ;READ ALL HEADERS
                        CONCLR= 176             ;CONTROLLER CLEAR (BIT 15 OF CS1)
                        SUBCLR= 177             ;SUBSYSTEM CLEAR (BIT 5 OF CS2)
                        INTR=   300             ;GENERATE INTERRUPT TO CPU

                        ;       DRIVER ISSUED SERVICE COMMANDS

                        DR.SEL= 001             ;DRIVE SELECT
                        DR.CLR= 005             ;DRIVE CLEAR
```

```
1191
1192                                .SBTTL  CONTROL AND   ATUS REGISTER 1 BITS
1193
1194            000001              GO=     BIT0                    ;GO BIT
1195            000100              IE=     BIT6                    ;INTERRUPT ENABLE
1196            000200              RDY=    BIT7                    ;CONTROLLER READY
1197            000400              BA16=   BIT8                    ;BUS ADDRESS BIT 16
1198            001000              BA17=   BIT9                    ;BUS ADDRESS BIT 17
1199            002000              CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
1200            004000              CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
12u1                                                                ;  DRIVE RESPONSE
1202            010000              CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1203            020000              SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
1204            040000              DI=     BIT14                   ;DRIVE INTERRUPT
1205            100000              CERR=   BIT15                   ;CONTROLLER ERROR
1206            100000              CCLR=   BIT15                   ;CONTROLLER CLEAR
1207
1208                                ;       THESE BIT DEFINITIONS ARE USED FOR ADDRESS
1209                                ;       THE HIGH BYTE OF RKCS1
1210
1211            000001              B.BA16= BIT0                    ;BUS ADDRESS BIT 16
1212            000002              B.BA17= BIT1                    ;BUS ADDRESS BIT 17
1213            000004              B.CDT=  BIT2                    ;CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
1214            000020              B.CFMT= BIT4                    ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1215
1216                                .SBTTL  CONTROL AND STATUS REGISTER 2 BITS
1217
1218            000007              DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
1219            000010              DESL=   BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1220            000010              RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1221            000020              BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
1222            000040              CLR=    BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
1223            000040              SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
1224            000100              IR=     BIT6                    ;INPUT READY
1225            000200              OR=     BIT7                    ;OUTPUT READY
1226            000400              UFE=    BIT8                    ;UNIT FIELD ERROR
1227            001000              MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1228            002000              PGE=    BIT10                   ;PROGRAMMING ERROR
1229            004000              NEM=    BIT11                   ;NON-EXISTENT MEMORY
1230            010000              NED=    BIT12                   ;NON-EXISTENT DRIVE
1231            020000              UPE=    BIT13                   ;UNIBUS PARITY ERROR
1232            040000              WCE=    BIT14                   ;WRITE CHECK ERROR
1233            100000              DLT=    BIT15                   ;DATA LATE ERROR
1234
1235                                .SBTTL  ERROR REGISTER BIT DEFINITION
1236
1237            000001              ILC=    BIT0                    ;ILLEGAL FUNCTION CODE
1238                                ;*ILF=  BIT0                    ;ILLEGAL FUNCTION CODE
1239            000002              SKI=    BIT1                    ;SEEK INCOMPLETE
1240            000004              ILF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
1241            000004              NXF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
1242            00001C              DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
1243            000020              FMTE=   BIT4                    ;FORMAT ERROR
1244            000040              DTYE=   BIT5                    ;DRIVE TYPE ERROR
1245            000100              ECH=    BIT6                    ;ECC HARD
1246            000200              BSE=    BIT7                    ;BAD SECTOR ERROR
```

```
1247    000400          HCRC=   BIT8                    ;HEADER CRC ERRROR
1248    000400          HVRC=   BIT8                    ;HEADER VRC ERRROR
1249    001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1250    002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
1251    004000          WLE=    BIT11                   ;WRITE LOCK ERROR
1252    010000          DTF=    BIT12                   ;DRIVE TIMING ERROR
1253    020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1254    040000          UNS=    BIT14                   ;DRIVE UNSAFE
1255    100000          DCK=    BIT15                   ;DATA CHECK
1256
1257                    .SBTTL  STATUS REGISTER BIT DEFINITION
1258
1259    000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1260                                                    ;  THIS BIT IS RESET)
1261    000004          OFST=   BIT2                    ;DRIVE OFFSET
1262    000010          ACLO=   BIT3                    ;AC LOW
1263    000020          SPDLSS= BIT4                    ;SPEED LOSS
1264    000020          DCLO=   BIT4                    ;DC LOW
1265    000040          DROT=   BIT5                    ;DRIVE OFF TRACK
1266    000100          VV=     BIT6                    ;VOLUME VALID
1267    000200          DRY=    BIT7                    ;DRIVE READY
1268    000200          DRDY=   BIT7                    ;DRIVE READY
1269    000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06, 1=RK07)
1270    004000          WRL=    BIT11                   ;WRITE LOCK
1271    020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
1272    040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
1273    100000          SVAL=   BIT15                   ;STATUS VALID
1274
1275                    .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION
1276
1277    000017          MESMSK= 17                      ;MESSAGE MASK
1278
1279    000020          PAT=    BIT4                    ;FORCE EVEN PARITY ON DRIVE BUS MESSAGE LINES
1280    000040          DMD=    BIT5                    ;DIAGNOSTIC MODE
1281    000100          MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1282    000200          MIND=   BIT7                    ;MAINTENANCE INDEX
1283    000400          MCLK=   BIT8                    ;MAINTENANCE CLOCK
1284    001000          MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1285    002000          MEWD=   BIT10                   ;MAINTENACNE ENCODED WRITE DATA
1286    004000          PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
1287    010000          PCD=    BIT12                   ;PRECOMPENSATION DELAY
1288    020000          ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
1289    040000          WRTGAT= BIT14                   ;WRITE GATE
1290    100000          RDGATE= BIT15                   ;READ GATE
1291
1292                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
1293
1294    000040          S.DRA=  BIT5                    ;DRIVE AVAILIABLE
1295    000100          S.VV=   BIT6                    ;VOLUME VALID
1296    000200          S.DRY=  BIT7                    ;DRIVE READY
1297    000400          S.TYPE= BIT8                    ;DRIVE TYPE
1298    001000          S.FORM= BIT9                    ;DRIVE FORMAT
1299    002000          S.OFF=  BIT10                   ;OFFSET
1300    004000          S.WRL=  BIT11                   ;WRITE LOCK
1301    010000          S.SPIN= BIT12                   ;SPINDLE ON
1302    020000          S.PIP=  BIT13                   ;POSITIONING IN PROGRESS
```

```
                        S.DSC=  BIT14                   ;DRIVE STATUS CHANGE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B

000040                  S.ICYL= BIT5                    ;ILLEGAL CYLINDER ADDRESS
000100                  S.ACLO= BIT6                    ;AC LOW
000200                  S.FLT=  BIT7                    ;DRIVE FAULT
000400                  S.ILF=  BIT8                    ;ILLEGAL FUNCTION
001000                  S.PAR=  BIT9                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
002000                  S.SKI=  BIT10                   ;SEEK INCOMPLETE
004000                  S.WLE=  BIT11                   ;WRITE LOCK ERROR
010000                  S.SPLS= BIT12                   ;SPEED LOSS
010000                  S.DCLO= BIT12                   ;DC LOW
020000                  S.DROT= BIT13                   ;DRIVE OFF TRACK
040000                  S.UNS=  BIT14                   ;DRIVE UNSAFE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A

000020                  S.XDOK= BIT4                    ;TRANSDUCER OK
000040                  S.HDHM= BIT5                    ;HEADS HOME
000100                  S.BRHM= BIT6                    ;BRUSHES HOME
000200                  S.DOOR= BIT7                    ;DOOR INTERLOCKED
000400                  S.CART= BIT8                    ;CARTRAGE INTERLOCK
001000                  S.SPOK= BIT9                    ;SPEED OK
002000                  S.FWD=  BIT10                   ;FORWARD
004000                  S.REV=  BIT11                   ;REVERSE
010000                  S.LOAD= BIT12                   ;HEADS LOADING
020000                  S.RTZ=  BIT13                   ;RETURN TO ZERO
040000                  S.UNLD= BIT14                   ;HEADS UNLOADING

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B

000020                  S.SECT= BIT4                    ;SECTOR ERROR
000040                  S.WCLK= BIT5                    ;WRITE CLOCK AND NO WRITE GATE
000100                  S.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
000200                  S.HDFL= BIT7                    ;HEAD FAULT
000400                  S.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
001000                  S.XERR= BIT9                    ;INDEX ERROR
002000                  S.DIB=  BIT10                   ;DIBIT ERROR
004000                  S.PLO=  BIT11                   ;PLO ERROR
010000                  S.NMOV= BIT12                   ;SEEK AND NO MOTION
020000                  S.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
040000                  S.BRKE= BIT14                   ;SERVO-BRAKE

                        .SBTTL  COMMON MASKS

000007                  M.DRV=  7                       ;DRIVE CODE
100000                  M.PAR=  BIT15                   ;PARITY
000003                  M.ID=   3                       ;BYTE ID
017760                  M.CDIF= 17760                   ;CYLINDER DIFFERENCE/OFFSET
017760                  M.CADD= 17760                   ;CYLINDER ADDRESS
077770                  M.SER=  77770                   ;DRIVE SERIAL NUMBER
000760                  M.SECT= 760                     ;SECTOR COUNT
007000                  M.HEAD= 7000                    ;HEAD DECODE

                        .SBTTL  DEFAULT DRIVE VALUES
```

```
1359
1360    000000          D.MNCL= 0               ;MINIMUM CYLINDER
1361    000000          D.MNTR= 0               ;MINIMUM TRACK
1362    000002          D.MXTR= 2               ;MAXIMUM TRACK
1363    000000          D.MNSC= 0               ;MINIMUM SECTOR
1364    000025          D.MXSC= 21.             ;MAXIMUM SECTOR
1365    000003          D.RATE= 3               ;READ/WRITE RATIO (5/3)
1366    000000          D.AWCK= 0               ;NO AUTOMATIC WRITE-CHECK AFTER EVERY WRITE
1367    000012          D.CERT= 10.             ;CORRECTABLE READ ERROR THRESHOLD
1368    000005          D.UERT= 5               ;UNCORRECTABLE READ ERROR THRESHOLD
1369    000005          D.SKET= 5               ;SEEK ERROR THRESHOLD
1370    077777          D.CTRH= 77777           ;COMMAND COUNT THRESHOLD
1371    177770          D.CTRL= 177770          ;  (MAXIMUM OF COMMANDS ISSUED TO DRIVE)
1372    077777          D.WTHI= 77777           ;WORDS TRANSFERRED THRESHOLD
1373    177770          D.WTLO= 177770          ;  (MAXIMUM NUMBER OF WORDS READ OR WRITTEN
1374                                            ;  ON PACK)
1375    000020          T.NER= 20               ;THRESHOLD OF OTHER DRIVE ERRORS
1376
1377                    .SBTTL  OFFSET VALUES
1378
1379    000060          OFFP12= 060             ;OFFSET +1200 MICRO-INCHES
1380    000040          OFFP8=  040             ;OFFSET +800 MICRO-INCHES
1381    000020          OFFP4=  020             ;OFFSET +400 MICRO-INCHES
1382    000220          OFFM4=  220             ;OFFSET -400 MICRO-INCHES
1383    000240          OFFM8=  240             ;OFFSET -800 MICRO-INCHES
1384    000260          OFFM12= 260             ;OFFSET -1200 MICRO-INCHES
1385
1386                    .SBTTL  DEFAULT CONTROLLER THRESHOLD
1387
1388    001000          T.DLT=  1000            ;THRESHOLD FOR DATA LATE
1389    000012          T.CONT= 10.             ;THRESHOLD FOR OTHER CONTROLLER ERRORS
```

```
1390                            .SBTTL  PARAMETER BLOCK ALLOCATION
1391
1392                            ;*      ************************************************
1393                            ;*    1 ! COMMAND                    ! DRIVE NO.      !  0
1394                            ;*    3 ! CYLINDER ADDRESS                            !  2
1395                            ;*    5 ! TRACK                      ! SECTOR         !  4
1396                            ;*    7 !BA16-17,FORMAT,DRV TYPE     ! OFFSET         !  6
1397                            ;*   11 ! BUS ADDRESS (LOW 16 BITS)                   ! 10
1398                            ;*   13 ! WORD COUNT (2'S COMPLEMENT)                 ! 12
1399                            ;*   15 ! PROGRAM DRIVE STATUS INFORMATION            ! 14
1400                            ;*   17 ! COMMAND AND STATUS REGISTER 1               ! 16
1401                            ;*   21 ! COMMAND AND STATUS REGISTER 2               ! 20
1402                            ;*   23 ! WORD COUNT REGISTER                         ! 22
1403                            ;*   25 ! BUS ADDRESS REGISTER                        ! 24
1404                            ;*   27 ! DESIRED TRACK AND SECTOR                    ! 26
1405                            ;*   31 ! DESIRED CYLINDER                            ! 30
1406                            ;*   33 ! ATTENTION SUMMARY AND DRIVE OFFSET          ! 32
1407                            ;*   35 ! ERROR REGISTER                              ! 34
1408                            ;*   37 ! STATUS REGISTER                             ! 36
1409                            ;*   41 ! MESSAGE LINE A STATUS BYTE 00               ! 40
1410                            ;*   43 ! MESSAGE LINE B STATUS BYTE 00               ! 42
1411                            ;*   45 ! MESSAGE LINE A STATUS BYTE 01               ! 44
1412                            ;*   47 ! MESSAGE LINE B STATUS BYTE 01               ! 46
1413                            ;*   51 ! MESSAGE LINE A STATUS BYTE 10               ! 50
1414                            ;*   53 ! MESSAGE LINE B STATUS BYTE 10               ! 52
1415                            ;*   55 ! MESSAGE LINE A STATUS BYTE 11               ! 54
1416                            ;*   57 ! MESSAGE LINE B STATUS BYTE 11               ! 56
1417                            ;*   61 ! ECC POSITION INFORMATION                    ! 60
1418                            ;*   63 ! ECC PATTERN INFORMATION                     ! 62
1419                            ;*      ************************************************
1420
1421                            .SBTTL  PARAMETERS PASSED TO THE DRIVER
1422
1423                            ;       THE FOLLOWING DEFINITIONS ARE USED TO PASS PARAMETERS
1424                            ;         TO THE RK06/07 DRIVER
1425
1426  000000                    P.DRVN= 0                       ;DRIVE NUMBER
1427  000001                    P.CMND= 1                       ;COMMAND
1428  000002                    P.CYLN= 2                       ;CYLINDER ADDRESS
1429  000004                    P.SECT= 4                       ;SECTOR
1430  000005                    P.TRCK= 5                       ;TRACK
1431  000006                    P.OFST= 6                       ;OFFSET
1432  000007                    P.CS1H= 7                       ;RKCS1 BITS 8-15
1433  000007                    P.BAHI= 7                       ;BUS ADDRESS (BITS 16 AND 17)
1434  000010                    P.BALO= 10                      ;BUS ADDRESS (BITS 0-15)
1435  000012                    P.WC=   12                      ;WORD COUNT (2'S COMPLEMENT)
1436  000014                    P.PRST= 14                      ;PROGRAM DRIVE STATUS INFORMATION
1437
1438                            .SBTTL  PROGRAM DEVICE STATUS REGISTER DEFINITION
1439
1440  000001                    DRVUSE= BIT0                    ;DRIVE IN USE
1441  000002                    DRVPOS= BIT1                    ;DRIVE POSITIONING
1442  000004                    DRVPDT= BIT2                    ;DRIVE POSITIONED FOR DATA TRANSFER
1443  000010                    UEXATT= BIT3                    ;UNEXPECTED ATTENTION
1444  000020                    DRVHRD= BIT4                    ;DRIVE HAS HARD ERROR
1445  000040                    DRVDSC= BIT5                    ;DRIVE STATUS CHANGE DID NOT CLEAR
```

```
1446            000100                  CMDTO=  BIT6                    ;NO TERMINATION TO COMMAND FOR AT
1447                                                                    ;  LEAST 1 SECOND
1448            000200                  W.WCK=  BIT7                    ;WRITE FOR WRITE WRITE CHECK
1449            000400                  NOCHK=  BIT8                    ;NO CHECK, DO NOT SET INTERRUPT ENABLE
1450            001000                  PBSVAL= BIT9                    ;PARAMETER STATUS WORDS VALID
1451                                                                    ;  (SET WHEN ERROR TERMINATION OR
1452                                                                    ;  READ STATUS COMMAND)
1453            002000                  DRPDRV= BIT10                   ;DROP DRIVE FROM TEST SEQUENCE
1454            004000                  NODSC=  BIT11                   ;ATTENTION SET BUT DCS AND FAULT RESET
1455            010000                  DRVSZD= BIT12                   ;DRIVE SEIZED BY OTHER PORT
1456            020000                  E.UNLD= BIT13                   ;DRIVE UNLOADED DUE TO ERROR
1457            040000                  Q.INIT= BIT14                   ;PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
1458            100000                  DTBAII= BIT15                   ;INHIBIT BUS ADDRESS INCREMENT
1459
1460                                    .SBTTL  PARAMETERS PASSED FROM DRIVER TO PROGRAM
1461
1462                                    ;       THE FOLLOWING DEFINITIONS ARE USED FOR REGISTER RETURNS
1463                                    ;         FROM THE DRIVER TO THE CALLING PROGRAM
1464
1465            000016                  P.CS1=  16                      ;COMMAND AND STATUS REGISTER 1
1466            000020                  P.CS2=  20                      ;COMMAND AND STATUS REGISTER 2
1467            000022                  P.WCR=  22                      ;WORD COUNT REGISTER
1468            000024                  P.BAR=  24                      ;BUS ADDPESS REGISTER
1469            000026                  P.DTS=  26                      ;DESIRED TRACK SECTOR REGISTER
1470            000030                  P.DCYL= 30                      ;DESIRED CYLINDER REGISTER
1471            000032                  P.ASOF= 32                      ;ATTENTION SUMMARY/OFFSET REGISTER
1472            000034                  P.ER=   34                      ;ERROR REGISTER
1473            000036                  P.DS=   36                      ;STATUS REGISTER
1474            000040                  P.A00=  40                      ;MESSAGE A STATUS BYTE 00
1475            000042                  P.B00=  42                      ;MESSAGE B STATUS BYTE 00
1476            000044                  P.A01=  44                      ;MESSAGE A STATUS BYTE 01
1477            000046                  P.B01=  46                      ;MESSAGE B STATUS BYTE 01
1478            000050                  P.A10=  50                      ;MESSGGE A STATUS BYTE 10
1479            000052                  P.B10=  52                      ;MESSAGE B STATUS BYTE 10
1480            000054                  P.A11=  54                      ;MESSAGE A STATUS BYTE 11
1481            000056                  P.B11=  56                      ;MESSAGE B STATUS BYTE 11
1482            000060                  P.EPOS= 60                      ;ECC POSITION INFORMATION
1483            000062                  P.EPAT= 62                      ;ECC PATTERN INFORMATION
1484
1485                                    .SBTTL  DRIVE SPECIFIC PARAMETERS
1486
1487            000064                  P.QLNK= 64                      ;QUEUE LINK
```

```
1488                                ;       THE FOLLOWING DEFINITIONS ARE USED FOR DRIVE PARAMETER ACCESS
1489
1490    000066                      P.MNCL= 66                      ;MINIMUM CYLINDER
1491    000070                      P.MXCL= 70                      ;MAXIMUM CYLINDER
1492    000072                      P.MNTR= 72                      ;MINIMUM TRACK
1493    000073                      P.MXTR= 73                      ;MAXIMUM TRACK
1494    000074                      P.MNSC= 74                      ;MINIMUM SECTOR
1495    000075                      P.MXSC= 75                      ;MAXIMUM SECTOR
1496    000076                      P.RATE= 76                      ;READ/WRITE RATIO
1497    000077                      P.AWCK= 77                      ;AUTOMATIC WRITE CHECK
1498    000100                      P.CERT= 100                     ;CORRECTABLE READ ERROR THRESHOLD
1499    000102                      P.UERT= 102                     ;UNCORRECTABLE READ ERROR THRESHOLD
1500    000104                      P.SKET= 104                     ;SEEK ERROR THRESHOLD
1501    000106                      P.MXCD= 106                     ;MAXIMUM COMMANDS ISSUED TO DRIVE
1502    000112                      P.MXWT= 112                     ;MAXIMUM NUMBER WORDS TRANSFERRED
1503    000116                      P.SMPL= 116                     ;SAMPLE COMPARE FLAG
1504    000117                      P.ECMP= 117                     ;ECC COMPARE FLAG
1505    000120                      P.IERC= 120                     ;INHIBIT ERROR CORRECTION
1506    000121                      P.DPAT= 121                     ;DATA PATTERN USED
1507
1508                                ;       THE FOLLOWING DEFINITIONS ARE USED IN STORING
1509                                ;         CURRENT RANDOMLY GENERATED COMMAND
1510
1511    000122                      P.RDPT= 122                     ;CURRENT RANDOMLY GENERATED DATA PATTERN
1512    000123                      P.RCMD= 123                     ;CURRENT RANDOMLY GENERATED COMMAND
1513    000124                      P.RCYL= 124                     ;CURRENT RANDOMLY GENERATED CYLINDER
1514    000126                      P.RSEC= 126                     ;CURRENT RANDOMLY GENERATED SECTOR
1515    000127                      P.RTRK= 127                     ;CURRENT RANDOMLY GENERATED TRACK
1516    000130                      P.RWC=  130                     ;CURRENT RANDOMLY GENERATED WORD COUNT
1517    000132                      P.RBAL= 132                     ;CURRENT RANDOMLY GENERATER BUS ADDRESS
1518    000134                      P.RBAH= 134
1519
1520                                ;       THE FOLLOWING DEFINITIONS ARE USED IN STORING
1521                                ;         LAST ISSUED
1522
1523    000135                      P.LCMD= 135                     ;LAST COMMAND
1524    000136                      P.LCYL= 136                     ;LAST CYLINDER
1525    000140                      P.LSEC= 140                     ;LAST SECTOR
1526    000141                      P.LTRK= 141                     ;LAST TRACK
1527    000142                      P.LWC=  142                     ;LAST WORD COUNT
1528    000144                      P.LDPT= 144                     ;LAST DATA PATTERN
1529    000145                      P.BUFF= 145                     ;BUFFER ALLOCATED
```

```
1530                            ;       THE FOLLOWING IS USED FOR THE RETRY SEQUENCE
1531
1532        000146              P.RECT= 146                 ;RETRY COUNT
1533        000147              P.RERD= 147                 ;RETRY INDICATION FOR DATA ERRORS
1534                                                        ;0      NOT IN RETRY SEQUENCE
1535                                                        ;1-20   REREAD AT CENTER LINE
1536                                                        ;21     OFFSET +400 MICRO-INCHES
1537                                                        ;22-23  REREAD OFFSET +400 MICRO-INCHES
1538                                                        ;24     OFFSET -400 MICRO-INCHES
1539                                                        ;25-26  REREAD OFFSET -400 MICRO-INCHES
1540                                                        ;27     OFFSET +800 MICRO-INCHES
1541                                                        ;30-31  REREAD OFFSET +800 MICRO-INCHES
1542                                                        ;32     OFFSET -800 MICRO-INCHES
1543                                                        ;33-34  REREAD OFFSET -800 MICRO-INCHES
1544                                                        ;35     OFFSET +1200 MICRO-INCHES
1545                                                        ;36-37  REREAD OFFSET +1200 MICRO-INCHES
1546                                                        ;40     OFFSET -1200 MICRO-INCHES
1547                                                        ;41-42  REREAD OFFSET -1200 MICRO-INCHES
1548
1549        000150              P.DSTT= 150                 ;DRIVE STATUS INFORMATION FOR ERROR PROCESSING
1550                                                        ;BIT 0  INITIAL DATA ERROR
1551                                                        ;BIT 1  DATA ERROR BEING PROCESSED
1552                                                        ;BIT 2  NON-DATA ERROR BEING PROCESSED
1553                                                        ;BIT 3  SECTOR IN ERROR PRINTED
1554                                                        ;BIT 4  READ HEADER ISSUED
1555                                                        ;BIT 6  WAITING FOR READ STATUS FOR
1556                                                        ;       POSITIONING TIME OUT
1557                                                        ;BIT 7  FIRST ERROR
1558                                                        ;BIT 11 RECALIBRATE HAS BEEN ISSUED
1559                                                        ;BIT 12 SEEK HAS BEEN ISSUED
1560                                                        ;BIT 13 OFFSET HAS BEEN ISSUED
1561                                                        ;BIT 14 BAD SECTOR PROCESSING FOR WRITE CHECK
1562                                                        ;BIT 15 ERROR ENQUEUED
1563
1564                            ;       THE FOLLOWING DEFINITIONS ARE USED IN STATISTICAL GATHERING
1565
1566        000152              P.NODR= 152                 ;NUMBER OF ORDERS (2 WORDS)
1567        000156              P.NWRT= 156                 ;NUMBER OR WORDS WRITTEN (3 WORDS)
1568        000164              P.NRD=  164                 ;NUMBER OF WORDS READ (3 WORDS)
1569        000172              P.SRRD= 172                 ;NUMBER SOFT ERRORS (REREAD CORRECTABLE)
1570        000174              P.SOFF= 174                 ;NUMBER SOFT ERRORS (OFFSET CORRECTABLE)
1571        000176              P.SECC= 176                 ;NUMBER SOFT ERRORS (ECC CORRECTABLE)
1572        000200              P.HARD= 200                 ;NUMBER OF HARD DATA ERRORS
1573        000202              P.NSKI= 202                 ;NUMBER OF SEEK INCOMPLETE
1574        000204              P.NOPI= 204                 ;NUMBER OF OPERATION INCOMPLETES
1575        000206              P.NER=  206                 ;NUMBER OF ALL OTHER ERRORS
```

```
1576                            ;       THE FOLLOWING IS USED FOR THE DRIVE ASSIGNMENT SEQUENCE
1577
1578        000210              P.ASSN= 210                     ;ASSIGNMENT COMMAND SEQUENCE
1579                                                            ;0      INITIAL STATE
1580                                                            ;BIT 0  DRIVE SERIAL NUMBER
1581                                                            ;BIT 1  START SPINDLE
1582                                                            ;BIT 2  PACK ACKNOWLEDGE
1583                                                            ;BIT 3  READ PACK SERIAL NUMBER
1584                                                            ;BIT 4  WRITE PACK COMMAND ISSUED
1585                                                            ;BIT 5  END OF PACK WRITE
1586                                                            ;BIT 6  RECAL
1587                                                            ;BIT 7  RECAL AND RETRY READ PACK
1588                                                            ;       SERIAL NUMBER
1589
1590        000211              P.SEEK= 211                     ;SEEK TO PREVIOUS CYLINDER FLAG
1591
1592                            ;       THE FOLLOWING LOCATION IS USED TO INDICATE ERRORS
1593
1594        000212              P.ERR=  212                     ;BIT 2  DATA COMPARISON ERROR
1595                                                            ;BIT 3  DRIVE ERROR NOT REPORTED
1596                                                            ;BIT 4  DRIVE SEIZED TIME OUT
1597                                                            ;BIT 5  DRIVE POSITIONING TIME OUT
1598                                                            ;BIT 6  ERROR WHILE ENQUEUED
1599                                                            ;BIT 7  ATTENTION WHEN DRIVE NOT IN USE
1600                                                            ;BIT 8  TIME OUT WHILE WAITING FOR HEADS
1601                                                            ;       TO LOAD AFTER ERROR
1602                                                            ;BIT 10 DRIVE NOT READY AFTER START SPINDLE
1603                                                            ;BIT 11 PACK ACKNOWLEDGE DID NOT SET VOLUME
1604                                                            ;       VALID
1605
1606        000214              P.ERHD= 214                     ;HEADER ADDRESS OF FIRST DATA MISCOMPARE (3 BYTES)
1607        000217              P.ERAD= 217                     ;OFFSET COUNT FROM HEADER ADDRESS
1608        000220              P.CMLG= 220                     ;COMPARISON LENGTH
1609        000222              P.BFCP= 222                     ;BUFFER COMPARE LENGTH
1610
1611                            ;       THE FOLLOWING LOCATIONS ARE USED FOR DRIVE SERIAL NUMBER,
1612                            ;         PACK SERIAL NUMBER, AND TRACK EXCLUSIONS
1613
1614        000224              P.SERL= 224                     ;DRIVE SERIAL NUMBER
1615        000226              P.PKSR= 226                     ;CARTRIAGE SERIAL NUMBER
1616        000232              P.EXAR= 232                     ;EXCLUDED AREA INFORMATION
```

```
                                        .SBTTL  TRAP CATCHER

                                                .=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
                                                .=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
                                                .=214
000214  000137  006642                          JMP     PARSRT          ;GET SYSTEM PARAMETERS AND START TESTING
000220  000137  065460                          JMP     ..DUMP          ;  *** MEMORY DUMP (DEBUG ONLY)
000224  000700                          ..LOW:  700                     ;START OF DUMP
000226  004600                          ..HIGH: PATTAB-2                ;LAST ADDRESS OF DUMP
                                                .=240
000240  000137  006662                          JMP     START1
                                                .=1000
                                        .SBTTL  APT PARAMETER BLOCK

                                        ;;**************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                        ;;**************************************************************
                                                .$X=.   ;;SAVE CURRENT LOCATION
                                                .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                                                200     ;;FOR APT START UP
                                                .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                                                $APTHDR ;;POINT TO APT HEADER BLOCK
                                                .=.$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                  000000  $CMTAG:                 ;;START OF COMMON TAGS
                                        .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
```

```
1715  001210      000                 $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1716  001211      000                 $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1717  001212  000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1718  001214  000000                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1719  001216  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1720                                  :*                      BITS 15-11=CPU TYPE
1721                                  :*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1722                                  :*                              11/70=06,PDQ=07,Q=10
1723                                  :*                      BIT 10=REAL TIME CLOCK
1724                                  :*                      BIT  9=FLOATING POINT PROCESSOR
1725                                  :*                      BIT  8=MEMORY MANAGEMENT
1726  001220      000                 $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1727  001221      000                 $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1728                                  :*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1729                                  :*                              900 NSEC CORE=001
1730                                  :*                              300 NSEC BIPOLAR=002
1731                                  :*                              500 NSEC MOS=003
1732  001222  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1733                                  :*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
1734  001224      000                 $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1735  001225      000                 $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1736  001226  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1737  001230      000                 $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1738  001231      000                 $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1739  001232  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1740  001234      000                 $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1741  001235      000                 $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1742  001236  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1743  001240  120210                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1744  001242  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1745  001244  177440                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1746  001246  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1747  001250  000000                  $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1748  001252  000000                  $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1749  001254  000000                  $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
1750  001256  000000                  $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
1751  001260  000000                  $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
1752  001262  000000                  $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1753  001264  000000                  $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1754  001266  000000                  $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1755  001270  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1756  001272  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1757  001274  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
1758  001276  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
1759  001300  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
1760  001302  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
1761  001304  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
1762  001306  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
1763  001310  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
1764  001312  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
1765
1766
1767  001314                          $ETEND:
1768
```

```
                                .SBTTL  QUEUE HEAD AND TAIL ALLOCATIONS

001314  000000  000000         AVAILQ: .WORD   0,0             ;HEAD AND TAIL OF AVAILABLE QUEUE
001320  000000  000000         BWAITQ: .WOR.   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 ADD..ESS
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 FA.L
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
                                                               ;  REG'STER 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
```

```
1825  001444  000000                  T.DB:   .WORD   0               ;TEMPORARY STORAGE FOR DATA BUFFER REGISTER
1826
1827                                  .SBTTL  DRIVER PARAMERTERS
1828
1829  001446  177440                  RKBAS:  .WORD   177440          ;ADDRESS OF RK611 UNIBUS ADDRESS BLOCK
1830  001450  000210                  RKVEC:  .WORD   210             ;ADDRESS OF R611 VECTOR
1831  001452  000240                  RKPRI:  .WORD   PR5             ;RK611 INTERRUPT PRIORITY
1832  001454  024400                  A.NORM: NORMAL                  ;ADDRESS OF NORMAL RETURN FROM DRIVER
1833  001456  025470                  A.ABNL: ABNORM                  ;ADDRESS OF ABNORMAL RETURN FROM DRIVER
1834  001460  026524                  A.CONT: CONTRL                  ;ADDRESS OF CONTROLLER ERROR RETURN
1835  001462  000000                  E.CONT: .WORD   0               ;CONTROLLER ERROR STATUS
1836                                                                  ;  THIS LOCATION IS CLEARED WHEN EVERY COMMAND
1837                                                                  ;  IS INITIATED.  IF A CONTROLLER ERROR
1838                                                                  ;  OCCURS THE FOLLOWING BIT ASSIGNMENT IS
1839                                                                  ;  USED:
1840                                                                  ;
1841          000001                  E.CCLR= BIT0                    ;CLEAR CONTROLLER DID NOT CLEAR ERROR
1842          000002                  E.NOAT= BIT1                    ;NO ATTENTION IN ATTENTION SUMMARY REG
1843          000004                  E.UATT= BIT2                    ;UNSOLICATED ATTENTION (SEQUENTIAL ONLY)
1844          000010                  E.UDAT= BIT3                    ;UNEXPECTED DATA TYPE ERROR
1845          000020                  E.CLAT= BIT4                    ;ATTENTION DID NOT RESET WITH CLEAR
1846          000040                  E.SCLR= BIT5                    ;SUBSYSTEM CLEAR DID NOT CLEAR DRIVE
1847                                                                  ;  ATTENTION
1848          000100                  E.ILLD= BIT6                    ;ILLEGAL DRIVER COMMAND
1849          000400                  E.DLT=  BIT8                    ;DATA LATE WHEN UNLOADING HEADER
1850          001000                  E.CERR= BIT9                    ;CONTROLLER ERROR DURING DRIVER SERVICING
1851          002000                  E.DPAR= BIT10                   ;DRIVE DETECTED PARITY ERROR
1852          040000                  E.CMTO= BIT14                   ;CONTROLLER COMMAND TIME OUT (QUEUED ONLY)
1853          100000                  E.MDS=  BIT15                   ;MULTIPLE DRIVE SELECT
1854
1855  001464  000000                  O.WAIT: .WORD   0               ;PARAMETER BLOCK OF THE DRIVE
1856                                                                  ;WAITING FOR COMMAND COMPLETION
1857  001466  000400                  W.MTIM: .WORD   400             ;LOOP COUNTER FOR MILISECOND SCAN OF DRIVE
1858  001470  000400                  W.MILI: .WORD   400             ;16 MILISECOND TIME FOR PROGRAM
1859                                                                  ;
1860                                                                  ;       CPU     VALUE
1861                                                                  ;       ---     -----
1862                                                                  ;
1863                                                                  ;       11/05   100
1864                                                                  ;       11/10
1865                                                                  ;       11/20
1866                                                                  ;       11/34
1867                                                                  ;       11/40
1868                                                                  ;       11/45   400
1869                                                                  ;       11/50
1870                                                                  ;       11/70
1871  001472  000300                  W.SEC:  .WORD   300             ;SECOND COUNT COUNT FOR ALL COMMANDS
1872                                                                  ;  EXCEPT START SPINDLE
1873  001474  003000                  W.8SEC: .WORD   3000            ;8 SECOND FOR DRIVE CYCLE DOWN
1874  001476  030000                  W.MIN:  .WORD   30000           ;MINUTE TIME FOR START SPINDLE
1875  001500  000000                  OPTFLG: .WORD   0               ;FLAG = 1 WHILE IN CMND OPTIMIZER
1876  001502  000000                  HDR.AD: .WORD   0               ;ADDRESS USED FOR READ ALL HEADERS
1877  001504  000000                  HDR.CT: .WORD   0               ;NUMBER OF HEADERS LEFT TO READ FOR READ
1878                                                                  ;  ALL HEADERS
1879  001506      000                 I.ISRL: .BYTE   0               ;INTERRUPT OR RELEASED COMMAND ISSUED
1880  001507      002     004     010 H.HEAD: .BYTE   2,4,10          ;HEAD DECODES
```

```
                                ;W.TIME: .BYTE   0              ;DRIVES BEING WATCH-DOG TIMED
                                ;O.OVER: .BYTE   -1             ;OVERLAPPED OPERATIONS
                                ;
                                ;       .SBTTL  TABLE OF INTERRUPT MASKS
                                ;
                                ;INTMSK: .BYTE   1              ;INTERRUPT FOR DRIVE 0
                                ;        .BYTE   2              ;INTERRUPT FOR DRIVE 1
                                ;        .BYTE   4              ;INTERRUPT FOR DRIVE 2
                                ;        .BYTE   10             ;INTERRUPT FOR DRIVE 3
                                ;        .BYTE   20             ;INTERRUPT FOR DRIVE 4
                                ;        .BYTE   40             ;INTERRUPT FOR DRIVE 5
                                ;        .BYTE   100            ;INTERRUPT FOR DRIVE 6
                                ;        .BYTE   200            ;INTERRUPT FOR DRIVE 7
                                ;
                                ;       .SBTTL  PARAMETER BLOCK TABLE
                                ;
001524  001736                  ;PBLKT:  PARM0                  ;ADDRESS OF DRIVE 0'S PARAMETER BLOCK
001526  002214                  ;        PARM1                  ;ADDRESS OF DRIVE 1'S PARAMETER BLOCK
001530  002472                  ;        PARM2                  ;ADDRESS OF DRIVE 2'S PARAMETER BLOCK
001532  002750                  ;        PARM3                  ;ADDRESS OF DRIVE 3'S PARAMETER BLOCK
001534  003226                  ;        PARM4                  ;ADDRESS OF DRIVE 4'S PARAMETER BLOCK
001536  003504                  ;        PARM5                  ;ADDRESS OF DRIVE 5'S PARAMETER BLOCK
001540  003762                  ;        PARM6                  ;ADDRESS OF DRIVE 6'S PARAMETER BLOCK
001542  004240                  ;        PARM7                  ;ADDRESS OF DRIVE 7'S PARAMETER BLOCK
                                ;
                                ;       .SBTTL  WATCH-DOG TIMER COUNTS
                                ;
001544  000000                  ;W.DRV:  .WORD   0              ;WATCH-DOG FOR DRIVE 0
001546  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 1
001550  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 2
001552  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 3
001554  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 4
001556  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 5
001560  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 6
001562  000000                  ;        .WORD   0              ;WATCH-DOG FOR DRIVE 7
```

```
                        .SBTTL  PERFORMANCE EXERCISER STATUS INFORMATION

C01564  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 SERCTOR 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

002036  000012          PARM0:  .BYTE   0               ;DRIVE 0
002037  000000                  .BYTE   0               ;COMMAND
002040  000000                  .WORD   0               ;CYLINDER ADDRESS
002042  000000                  .BYTE   0               ;SECTOR ADDRESS
002043  000000                  .BYTE   0               ;TRACK ADDRESS
002044  000000                  .BYTE   0               ;OFFSET VALUE
002045  000000                  .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
002046  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
002050  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
002052  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
002054  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
002056  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
002060  000000                  .WORD   0               ;WORD COUNT REGISTER
002062  000000                  .WORD   0               ;BUS ADDRESS REGISTER
002064  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
002066  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
002070  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
002072  000000                  .WORD   0               ;ERROR REGISTER
002074  000000                  .WORD   0               ;STATUS REGISTER
002076  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
```

```
2025  002070  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2026  002072     000                        .BYTE   0
2027  002073     000                        .BYTE   0               ;LAST COMMAND
2028  002074  000000                        .WORD   0               ;LAST CYLINDER
2029  002076     000                        .BYTE   0               ;LAST SECTOR
2030  002077     000                        .BYTE   0               ;LAST TRACK
2031  002100  000000                        .WORD   0               ;LAST WORD COUNT
2032  002102     000                        .BYTE   0               ;LAST DATA PATTERN USED
2033  002103     000                        .BYTE   0               ;BUFFER ALLOCATED
2034  002104     000                        .BYTE   0               ;RETRY COUNT
2035  002105     000                        .BYTE   0               ;REREAD STATUS
2036  002106  000000                        .WORD   0               ;DRIVE STATUS FLAGS
2037  002110  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
2038  002114  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2039  002122  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
2040  002130  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2041  002132  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2042  002134  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2043  002136  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
2044  002140  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2045  002142  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2046  002144  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2047  002146     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2048  002147     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2049  002150  000000                        .WORD   0               ;ERROR STATUS INFORMATION
2050  002152  000000                        .WORD   C               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2051  002154     000                        .BYTE   0
2052  002155     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2053  002156  000000                        .WORD   0               ;COMPARISON LENGTH
2054  002160  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
2055  002162  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
2056  002164  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
2057  002170  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
2058  002174  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
2059  002200  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
2060  002204  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
2061  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
```

```
 2118  002346  000000                          .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
 2119  002350     000                          .BYTE   0
 2120  002351     000                          .BYTE   0               ;LAST COMMAND
 2121  002352  000000                          .WORD   0               ;LAST CYLINDER
 2122  002354     000                          .BYTE   0               ;LAST SECTOR
 2123  002355     000                          .BYTE   0               ;LAST TRACK
 2124  002356  000000                          .WORD   0               ;LAST WORD COUNT
 2125  002360     000                          .BYTE   0               ;LAST DATA PATTERN USED
 2126  002361     000                          .BYTE   0               ;BUFFER ALLOCATED
 2127  002362     000                          .BYTE   0               ;RETRY COUNT
 2128  002363     000                          .BYTE   0               ;REREAD STATUS
 2129  002364  000000                          .WORD   0               ;DRIVE STATUS FLAGS
 2130  002366  000000  000000                  .WORD   0,0             ;NUMBER OF ORDERS
 2131  002372  000000  000000  000000          .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
 2132  002400  000000  000000  000000          .WORD   0,0,0           ;NUMBER OF WORDS READ
 2133  002406  000000                          .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
 2134  002410  000000                          .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
 2135  002412  000000                          .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
 2136  002414  000000                          .WORD   0               ;NUMBER OF HARD DATA ERRORS
 2137  002416  000000                          .WORD   0               ;NUMBER OF SEEK INCOMPLETES
 2138  002420  000000                          .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
 2139  002422  000000                          .WORD   0               ;NUMBER OF ALL OTHER ERRORS
 2140  002424     000                          .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
 2141  002425     000                          .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
 2142  002426  000000                          .WORD   0               ;ERROR STATUS INFORMATION
 2143  002430  000000                          .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
 2144  002432     000                          .BYTE   0
 2145  002433     000                          .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
 2146  002434  000000                          .WORD   0               ;COMPARISON LENGTH
 2147  002436  000000                          .WORD   0               ;BUFFER COMPARISON LENGTH
 2148  002440  007777                          .WORD   007777          ;DRIVE SERIAL NUMBER
 2149  002442  000000  000000                  .WORD   0,0             ;CARIAGE SERIAL NUMBER
 2150  002446  000000  000000                  .WORD   0,0             ;FIRST EXCLUSION AREA
 2151  002452  000000  000000                  .WORD   0,0             ;SECOND EXCLUSION AREA
 2152  002456  000000  000000                  .WORD   0,0             ;THIRD EXCLUSION AREA
 2153  002462  000000  000000                  .WORD   0,0             ;FOURTH EXCLUSION AREA
 2154  002466  000000  000000                  .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
                                .SBTTL  PARAMETER BLOCK FOR DRIVE 2

002472  002          PARM2:  .BYTE   2               ;DRIVE 2
002473  000                  .BYTE   0               ;COMMAND
002474  000000               .WORD   0               ;CYLINDER ADDRESS
002476  000                  .BYTE   0               ;SECTOR ADDRESS
002477  000                  .BYTE   0               ;TRACK ADDRESS
002500  000                  .BYTE   0               ;OFFSET VALUE
002501  000                  .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
002502  000000               .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
002504  000000               .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
002506  000000               .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
002510  000000               .WORD   0               ;COMMAND AND STATUS REGISTER 1
002512  000000               .WORD   0               ;COMMAND AND STATUS REGISTER 2
002514  000000               .WORD   0               ;WORD COUNT REGISTER
002516  000000               .WORD   0               ;BUS ADDRESS REGISTER
002520  000000               .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
002522  000000               .WORD   0               ;DESIRED CYLINDER REGISTER
002524  000000               .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
002526  000000               .WORD   0               ;ERROR REGISTER
002530  000000               .WORD   0               ;STATUS REGISTER
002532  000000               .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
002534  000000               .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
002536  000000               .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
002540  000000               .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
002542  000000               .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
002544  000000               .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
002546  000000               .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
002550  000000               .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
002552  000000               .WORD   0               ;ECC POSITION INFORMATION
002554  000000               .WORD   0               ;ECC PATTERN INFORMATION
002556  000000               .WORD   0               ;QUEUE LINK
002560  000000               .WORD   0               ;MINIMUM CYLINDER
002562  000632               .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
002564  000                  .BYTE   0               ;MINIMUM TRACK
002565  002                  .BYTE   2               ;MAXIMUM TRACK
002566  000                  .BYTE   0               ;MINIMUM SECTOR
002567  025                  .BYTE   21.             ;MAXIMUM SECTOR
002570  003                  .BYTE   3               ;READ/WRITE RATIO
002571  000                  .BYTE   0               ;AUTOMATIC WRITE CHECK
002572  000012               .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
002574  000005               .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
002576  000005               .WORD   5               ;SEEK ERROR THRESHOLD
002600  177770  177777       .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
002604  177770  177777       .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
002610  000                  .BYTE   0               ;SAMPLE COMPARE FLAG
002611  000                  .BYTE   0               ;ECC COMPARE FLAG
002612  000                  .BYTE   0               ;INHIBIT ERROR CORRECTION
002613  000                  .BYTE   0               ;DATA PATTERN USED
002614  000                  .BYTE   0               ;CURRENT GENERATED DATA PATTERN
002615  000                  .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
002616  000000               .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
002620  000                  .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
002621  000                  .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
002622  000000               .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2211  002624  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2212  002626     000                        .BYTE   0
2213  002627     000                        .BYTE   0               ;LAST COMMAND
2214  002630  000000                        .WORD   0               ;LAST CYLINDER
2215  002632     000                        .BYTE   0               ;LAST SECTOR
2216  002633     000                        .BYTE   0               ;LAST TRACK
2217  002634  000000                        .WORD   0               ;LAST WORD COUNT
2218  002636     000                        .BYTE   0               ;LAST DATA PATTERN USED
2219  002637     000                        .BYTE   0               ;BUFFER ALLOCATED
2220  002640     000                        .BYTE   0               ;RETRY COUNT
2221  002641     000                        .BYTE   0               ;REREAD STATUS
2222  002642  000000                        .WORD   0               ;DRIVE STATUS FLAGS
2223  002644  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
2224  002650  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2225  002656  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
2226  002664  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2227  002666  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2228  002670  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2229  002672  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
2230  002674  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2231  002676  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2232  002700  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2233  002702     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2234  002703     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2235  002704  000000                        .WORD   0               ;ERROR STATUS INFORMATION
2236  002706  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2237  002710     000                        .BYTE   0
2238  002711     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2239  002712  000000                        .WORD   0               ;COMPARISON LENGTH
2240  002714  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
2241  002716  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
2242  002720  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
2243  002724  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
2244  002730  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
2245  002734  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
2246  002740  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
2247  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
```

```
 2304  003102  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
 2305  003104     000                        .BYTE   0
 2306  003105     000                        .BYTE   0               ;LAST COMMAND
 2307  003106  000000                        .WORD   0               ;LAST CYLINDER
 2308  003110     000                        .BYTE   0               ;LAST SECTOR
 2309  003111     000                        .BYTE   0               ;LAST TRACK
 2310  003112  000000                        .WORD   0               ;LAST WORD COUNT
 2311  003114     000                        .BYTE   0               ;LAST DATA PATTERN USED
 2312  003115     000                        .BYTE   0               ;BUFFER ALLOCATED
 2313  003116     000                        .BYTE   0               ;RETRY COUNT
 2314  003117     000                        .BYTE   0               ;REREAD STATUS
 2315  003120  000000                        .WORD   0               ;DRIVE STATUS FLAGS
 2316  003122  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
 2317  003126  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
 2318  003134  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
 2319  003142  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
 2320  003144  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
 2321  003146  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
 2322  003150  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
 2323  003152  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
 2324  003154  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
 2325  003156  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
 2326  003160     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
 2327  003161     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
 2328  003162  000000                        .WORD   0               ;ERROR STATUS INFORMATION
 2329  003164  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
 2330  003166     000                        .BYTE   0
 2331  003167     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
 2332  003170  000000                        .WORD   0               ;COMPARISON LENGTH
 2333  003172  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
 2334  003174  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
 2335  003176  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
 2336  003202  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
 2337  003206  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
 2338  003212  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
 2339  003216  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
 2340  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
```

```
 2397  003360  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
 2398  003362     000                        .BYTE   0
 2399  003363     000                        .BYTE   0               ;LAST COMMAND
 2400  003364  000000                        .WORD   0               ;LAST CYLINDER
 2401  003366     000                        .BYTE   0               ;LAST SECTOR
 2402  003367     000                        .BYTE   0               ;LAST TRACK
 2403  003370  000000                        .WORD   0               ;LAST WORD COUNT
 2404  003372     000                        .BYTE   0               ;LAST DATA PATTERN USED
 2405  003373     000                        .BYTE   0               ;BUFFER ALLOCATED
 2406  003374     000                        .BYTE   0               ;RETRY COUNT
 2407  003375     000                        .BYTE   0               ;REREAD STATUS
 2408  003376  000000                        .WORD   0               ;DRIVE STATUS FLAGS
 2409  003400  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
 2410  003404  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
 2411  003412  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
 2412  003420  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE;
 2413  003422  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
 2414  003424  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
 2415  003426  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
 2416  003430  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
 2417  003432  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
 2418  003434  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
 2419  003436     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
 2420  003437     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
 2421  003440  000000                        .WORD   0               ;ERROR STATUS INFORMATION
 2422  003442  00000C                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
 2423  003444     000                        .BYTE   0
 2424  003445     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
 2425  003446  000000                        .WORD   0               ;COMPARISON LENGTH
 2426  003450  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
 2427  003452  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
 2428  003454  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
 2429  003460  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
 2430  003464  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
 2431  003470  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
 2432  003474  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
 2433  003500  000000  000000                .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
                                .SBTTL  PARAMETER BLOCK FOR DRIVE 5

003504  005             PARM5:  .BYTE   5               ;DRIVE 5
003505  000                     .BYTE   0               ;COMMAND
003506  000000                  .WORD   0               ;CYLINDER ADDRESS
003510  000                     .BYTE   0               ;SECTOR ADDRESS
003511  000                     .BYTE   0               ;TRACK ADDRESS
003512  000                     .BYTE   0               ;OFFSET VALUE
003513  000                     .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
003514  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
003516  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
003520  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
003522  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
003524  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
003526  000000                  .WORD   0               ;WORD COUNT REGISTER
003530  000000                  .WORD   0               ;BUS ADDRESS REGISTER
003532  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
003534  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
003536  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
003540  000000                  .WORD   0               ;ERROR REGISTER
003542  000000                  .WORD   0               ;STATUS REGISTER
003544  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
003546  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
003550  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
003552  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
003554  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
003556  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
003560  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
003562  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
003564  000000                  .WORD   0               ;ECC POSITION INFORMATION
003566  000000                  .WORD   0               ;ECC PATTERN INFORMATION
003570  000000                  .WORD   0               ;QUEUE LINK
003572  000000                  .WORD   0               ;MINIMUM CYLINDER
003574  000632                  .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
003576  000                     .BYTE   0               ;MINIMUM TRACK
003577  002                     .BYTE   2               ;MAXIMUM TRACK
003600  000                     .BYTE   0               ;MINIMUM SECTOR
003601  025                     .BYTE   21.             ;MAXIMUM SECTOR
003602  003                     .BYTE   3               ;READ/WRITE RATIO
003603  000                     .BYTE   0               ;AUTOMATIC WRITE CHECK
003604  000012                  .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
003606  000005                  .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
003610  000005                  .WORD   5               ;SEEK ERROR THRESHOLD
003612  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
003616  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
003622  000                     .BYTE   0               ;SAMPLE COMPARE FLAG
003623  000                     .BYTE   0               ;ECC COMPARE FLAG
003624  000                     .BYTE   0               ;INHIBIT ERROR CORRECTION
003625  000                     .BYTE   0               ;DATA PATTERN USED
003626  000                     .BYTE   0               ;CURRENT GENERATED DATA PATTERN
003627  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
003630  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
003632  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
003633  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
003634  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2490  003636  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2491  003640     000                  .BYTE   0
2492  003641     000                  .BYTE   0               ;LAST COMMAND
2493  003642  000000                  .WORD   0               ;LAST CYLINDER
2494  003644     000                  .BYTE   0               ;LAST SECTOR
2495  003645     000                  .BYTE   0               ;LAST TRACK
2496  003646  000000                  .WORD   0               ;LAST WORD COUNT
2497  003650     000                  .BYTE   0               ;LAST DATA PATTERN USED
2498  003651     000                  .BYTE   0               ;BUFFER ALLOCATED
2499  003652     000                  .BYTE   0               ;RETRY COUNT
2500  003653     000                  .BYTE   0               ;REREAD STATUS
2501  003654  000000                  .WORD   0               ;DRIVE STATUS FLAGS
2502  003656  000000  000000          .WORD   0,0             ;NUMBER OF ORDERS
2503  003662  000000  000000  000000  .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2504  003670  000000  000000  000000  .WORD   0,0,0           ;NUMBER OF WORDS READ
2505  003676  000000                  .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2506  003700  000000                  .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2507  003702  000000                  .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2508  003704  000000                  .WORD   0               ;NUMBER OF HARD DATA ERRORS
2509  003706  000000                  .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2510  003710  000000                  .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2511  003712  000000                  .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2512  003714     000                  .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2513  003715     000                  .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2514  003716  000000                  .WORD   0               ;ERROR STATUS INFORMATION
2515  003720  000000                  .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2516  003722     000                  .BYTE   0
2517  003723     000                  .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2518  003724  000000                  .WORD   0               ;COMPARISON LENGTH
2519  003726  000000                  .WORD   0               ;BUFFER COMPARISON LENGTH
2520  003730  007777                  .WORD   007777          ;DRIVE SERIAL NUMBER
2521  003732  000000  000000          .WORD   0,0             ;CARIAGE SERIAL NUMBER
2522  003736  000000  000000          .WORD   0,0             ;FIRST EXCLUSION AREA
2523  003742  000000  000000          .WORD   0,0             ;SECOND EXCLUSION AREA
2524  003746  000000  000000          .WORD   0,0             ;THIRD EXCLUSION AREA
2525  003752  000000  000000          .WORD   0,0             ;FOURTH EXCLUSION AREA
2526  003756  000000  000000          .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
2527
2528                                    .SBTTL  PARAMETER BLOCK FOR DRIVE 6
2529
2530  003762    006                     PARM6:  .BYTE   6               ;DRIVE 6
2531  003763    000                             .BYTE   0               ;COMMAND
2532  003764  000000                            .WORD   0               ;CYLINDER ADDRESS
2533  003766    000                             .BYTE   0               ;SECTOR ADDRESS
2534  003767    000                             .BYTE   0               ;TRACK ADDRESS
2535  003770    000                             .BYTE   0               ;OFFSET VALUE
2536  003771    000                             .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
2537  003772  000000                            .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
2538  003774  000000                            .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
2539  003776  000000                            .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
2540  004000  000000                            .WORD   0               ;COMMAND AND STATUS REGISTER 1
2541  004002  000000                            .WORD   0               ;COMMAND AND STATUS REGISTER 2
2542  004004  000000                            .WORD   0               ;WORD COUNT REGISTER
2543  004006  000000                            .WORD   0               ;BUS ADDRESS REGISTER
2544  004010  000000                            .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
2545  004012  000000                            .WORD   0               ;DESIRED CYLINDER REGISTER
2546  004014  000000                            .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
2547  004016  000000                            .WORD   0               ;ERROR REGISTER
2548  004020  000000                            .WORD   0               ;STATUS REGISTER
2549  004022  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
2550  004024  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
2551  004026  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
2552  004030  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
2553  004032  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
2554  004034  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
2555  004036  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
2556  004040  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
2557  004042  000000                            .WORD   0               ;ECC POSITION INFORMATION
2558  004044  000000                            .WORD   0               ;ECC PATTERN INFORMATION
2559  004046  000000                            .WORD   0               ;QUEUE LINK
2560  004050  000000                            .WORD   0               ;MINIMUM CYLINDER
2561  004052  000632                            .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
2562  004054    000                             .BYTE   0               ;MINIMUM TRACK
2563  004055    002                             .BYTE   2               ;MAXIMUM TRACK
2564  004056    000                             .BYTE   0               ;MINIMUM SECTOR
2565  004057    025                             .BYTE   21.             ;MAXIMUM SECTOR
2566  004060    003                             .BYTE   3               ;READ/WRITE RATIO
2567  004061    000                             .BYTE   0               ;AUTOMATIC WRITE CHECK
2568  004062  000012                            .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
2569  004064  000005                            .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
2570  004066  000005                            .WORD   5               ;SEEK ERROR THRESHOLD
2571  004070  177770  177777                  .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
2572  004074  177770  177777                  .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
2573  004100    000                             .BYTE   0               ;SAMPLE COMPARE FLAG
2574  004101    000                             .BYTE   0               ;ECC COMPARE FLAG
2575  004102    000                             .BYTE   0               ;INHIBIT ERROR CORRECTION
2576  004103    000                             .BYTE   0               ;DATA PATTERN USED
2577  004104    000                             .BYTE   0               ;CURRENT GENERATED DATA PATTERN
2578  004105    000                             .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
2579  004106  000000                            .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
2580  004110    000                             .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
2581  004111    000                             .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
2582  004112  000000                            .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
 2583  004114  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
 2584  004116     000                        .BYTE   0
 2585  004117     000                        .BYTE   0               ;LAST COMMAND
 2586  004120  000000                        .WORD   0               ;LAST CYLINDER
 2587  004122     000                        .BYTE   0               ;LAST SECTOR
 2588  004123     000                        .BYTE   0               ;LAST TRACK
 2589  004124  000000                        .WORD   0               ;LAST WORD COUNT
 2590  004126     000                        .BYTE   0               ;LAST DATA PATTERN USED
 2591  004127     000                        .BYTE   0               ;BUFFER ALLOCATED
 2592  004130     000                        .BYTE   0               ;RETRY COUNT
 2593  004131     000                        .BYTE   0               ;REREAD STATUS
 2594  004132  000000                        .WORD   0               ;DRIVE STATUS FLAGS
 2595  004134  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
 2596  004140  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
 2597  004146  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
 2598  004154  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
 2599  004156  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
 2600  004160  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
 2601  004162  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
 2602  004164  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
 2603  004166  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
 2604  004170  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
 2605  00417?     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
 2606  00417?     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
 2607  004174  000000                        .WORD   0               ;ERROR STATUS INFORMATION
 2608  004176  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
 2609  004200     000                        .BYTE   0
 2610  004201     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
 2611  004202  000000                        .WORD   0               ;COMPARISON LENGTH
 2612  004204  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
 2613  004206  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
 2614  004210  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
 2615  004214  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
 2616  004220  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
 2617  004224  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
 2618  004230  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
 2619  004234  000000  000000                .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
                                .SBTTL  PARAMETER BLOCK FOR DRIVE 7

004240  007             PARM7:  .BYTE   7               ;DRIVE 7
004241  000                     .BYTE   0               ;COMMAND
004242  000000                  .WORD   0               ;CYLINDER ADDRESS
004244  000                     .BYTE   0               ;SECTOR ADDRESS
004245  000                     .BYTE   0               ;TRACK ADDRESS
004246  000                     .BYTE   0               ;OFFSET VALUE
004247  000                     .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
004250  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
004252  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
004254  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
004256  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
004260  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
004262  000000                  .WORD   0               ;WORD COUNT REGISTER
004264  000000                  .WORD   0               ;BUS ADDRESS REGISTER
004266  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
004270  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
004272  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
004274  000000                  .WORD   0               ;ERROR REGISTER
004276  000000                  .WORD   0               ;STATUS REGISTER
004300  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
004302  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
004304  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
004306  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
004310  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
004312  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
004314  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
004316  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
004320  000000                  .WORD   0               ;ECC POSITION INFORMATION
004322  000000                  .WORD   0               ;ECC PATTERN INFORMATION
004324  000000                  .WORD   0               ;QUEUE LINK
004326  000000                  .WORD   0               ;MINIMUM CYLINDER
004330  000632                  .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
004332  000                     .BYTE   0               ;MINIMUM TRACK
004333  002                     .BYTE   2               ;MAXIMUM TRACK
004334  000                     .BYTE   0               ;MINIMUM SECTOR
004335  025                     .BYTE   21.             ;MAXIMUM SECTOR
004336  003                     .BYTE   3               ;READ/WRITE RATIO
004337  000                     .BYTE   0               ;AUTOMATIC WRITE CHECK
004340  000012                  .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
004342  000005                  .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
004344  000005                  .WORD   5               ;SEEK ERROR THRESHOLD
004346  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
004352  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
004356  000                     .BYTE   0               ;SAMPLE COMPARE FLAG
004357  000                     .BYTE   0               ;ECC COMPARE FLAG
004360  000                     .BYTE   0               ;INHIBIT ERROR CORRECTION
004361  000                     .BYTE   0               ;DATA PATTERN USED
004362  000                     .BYTE   0               ;CURRENT GENERATED DATA PATTERN
004363  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
004364  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
004366  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
004367  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
004370  0C0000                  .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
 2676  004372  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
 2677  004374     000                        .BYTE   0
 2678  004375     000                        .BYTE   0               ;LAST COMMAND
 2679  004376  000000                        .WORD   0               ;LAST CYLINDER
 2680  004400     000                        .BYTE   0               ;LAST SECTOR
 2681  004401     000                        .BYTE   0               ;LAST TRACK
 2682  004402  000000                        .WORD   0               ;LAST WORD COUNT
 2683  004404     000                        .BYTE   0               ;LAST DATA PATTERN USED
 2684  004405     000                        .BYTE   0               ;BUFFER ALLOCATED
 2685  004406     000                        .BYTE   0               ;RETRY COUNT
 2686  004407     000                        .BYTE   0               ;REREAD STATUS
 2687  004410  000000                        .WORD   0               ;DRIVE STATUS FLAGS
 2688  004412  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
 2689  004416  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
 2690  004424  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
 2691  004432  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
 2692  004434  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
 2693  004436  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
 2694  004440  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
 2695  004442  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
 2696  004444  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
 2697  004446  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
 2698  004450     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
 2699  004451     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
 2700  004452  000000                        .WORD   0               ;ERROR STATUS INFORMATION
 2701  004454  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
 2702  004456     000                        .BYTE   0
 2703  004457     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
 2704  004460  000000                        .WORD   0               ;COMPARISON LENGTH
 2705  004462  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
 2706  004464  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
 2707  004466  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
 2708  004472  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
 2709  004476  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
 2710  004502  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
 2711  004506  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
 2712  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

                                PATTAB: PAT0
                                        PAT1
                                        PAT2
                                        PAT3
                                        PAT4
                                        PAT5
                                        PAT6
                                        PAT7
                                        PAT10
                                        PAT11
                                        PAT12
                                        PAT13
                                        PAT14
                                        PAT15
                                        PAT16
                                        PAT17

                                PAT0:   .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333
                                        .WORD   165555
                                        .WORD   133333

                                PAT1:   .WORD   026455
                                        .WORD   151322
                                        .WORD   026455
                                        .WORD   026455
```

```
 2802  004752  151322                    .WORD   151322
 2803  004754  151322                    .WORD   151322
 2804  004756  026455                    .WORD   026455
 2805  004760  026455                    .WORD   026455
 2806  004762  026455                    .WORD   026455
 2807  004764  151322                    .WORD   151322
 2808  004766  151322                    .WORD   151322
 2809  004770  151322                    .WORD   151322
 2810  004772  026455                    .WORD   026455
 2811  004774  026455                    .WORD   026455
 2812  004776  026455                    .WORD   026455
 2813  005000  026455                    .WORD   026455
 2814  005002  151322                    .WORD   151322
 2815  005004  151322                    .WORD   151322
 2816  005006  151322                    .WORD   151322
 2817  005010  151322                    .WORD   151322
 2818  005012  026455                    .WORD   026455
 2819  005014  026455                    .WORD   026455
 2820  005016  026455                    .WORD   026455
 2821  005020  151322                    .WORD   151322
 2822  005022  151322                    .WORD   151322
 2823  005024  151322                    .WORD   151322
 2824  005026  026455                    .WORD   026455
 2825  005030  026455                    .WORD   026455
 2826  005032  151322                    .WORD   151322
 2827  005034  151322                    .WORD   151322
 2828  005036  026455                    .WORD   026455
 2829  005040  151322                    .WORD   151322
 2830
 2831  005042  007417            PAT2:   .WORD   007417
 2832  005044  170360                    .WORD   170360
 2833  005046  007417                    .WORD   007417
 2834  005050  007417                    .WORD   007417
 2835  005052  170360                    .WORD   170360
 2836  005054  170360                    .WORD   170360
 2837  005056  007417                    .WORD   007417
 2838  005060  007417                    .WORD   007417
 2839  005062  007417                    .WORD   007417
 2840  005064  170360                    .WORD   170360
 2841  005066  170360                    .WORD   170360
 2842  005070  170360                    .WORD   170360
 2843  005072  007417                    .WORD   007417
 2844  005074  007417                    .WORD   007417
 2845  005076  007417                    .WORD   007417
 2846  005100  007417                    .WORD   007417
 2847  005102  170360                    .WORD   170360
 2848  005104  170360                    .WORD   170360
 2849  005106  170360                    .WORD   170360
 2850  005110  170360                    .WORD   170360
 2851  005112  007417                    .WORD   007417
 2852  005114  007417                    .WORD   007417
 2853  005116  007417                    .WORD   007417
 2854  005120  170360                    .WORD   170360
 2855  005122  170360                    .WORD   170360
 2856  005124  170360                    .WORD   170360
 2857  005126  007417                    .WORD   007417
```

```
 2858  005130  007417                    .WORD   007417
 2859  005132  170360                    .WORD   170360
 2860  005134  170360                    .WORD   170360
 2861  005136  007417                    .WORD   007417
 2862  005140  170360                    .WORD   170360
 2863
 2864  005142  052525              PAT3: .WORD   052525
 2865  005144  125252                    .WORD   125252
 2866  005146  052525                    .WORD   052525
 2867  005150  052525                    .WORD   052525
 2868  005152  125252                    .WORD   125252
 2869  005154  125252                    .WORD   125252
 2870  005156  052525                    .WORD   052525
 2871  005160  052525                    .WORD   052525
 2872  005162  052525                    .WORD   052525
 2873  005164  125252                    .WORD   125252
 2874  005166  125252                    .WORD   125252
 2875  005170  125252                    .WORD   125252
 2876  005172  052525                    .WORD   052525
 2877  005174  052525                    .WORD   052525
 2878  005176  052525                    .WORD   052525
 2879  005200  052525                    .WORD   052525
 2880  005202  125252                    .WORD   125252
 2881  005204  125252                    .WORD   125252
 2882  005206  125252                    .WORD   125252
 2883  005210  125252                    .WORD   125252
 2884  005212  052525                    .WORD   052525
 2885  005214  052525                    .WORD   052525
 2886  005216  052525                    .WORD   052525
 2887  005220  125252                    .WORD   125252
 2888  005222  125252                    .WORD   125252
 2889  005224  125252                    .WORD   125252
 2890  005226  052525                    .WORD   052525
 2891  005230  052525                    .WORD   052525
 2892  005232  125252                    .WORD   125252
 2893  005234  125252                    .WORD   125252
 2894  005236  052525                    .WORD   052525
 2895  005240  125252                    .WORD   125252
 2896
 2897  005242  000000              PAT4: .WORD   000000
 2898  005244  010421                    .WORD   010421
 2899  005246  021042                    .WORD   021042
 2900  005250  031463                    .WORD   031463
 2901  005252  042104                    .WORD   042104
 2902  005254  052525                    .WORD   052525
 2903  005256  063146                    .WORD   063146
 2904  005260  073567                    .WORD   073567
 2905  005262  104210                    .WORD   104210
 2906  005264  114631                    .WORD   114631
 2907  005266  125252                    .WORD   125252
 2908  005270  135673                    .WORD   135673
 2909  005272  146314                    .WORD   146314
 2910  005274  156735                    .WORD   156735
 2911  005276  167356                    .WORD   167356
 2912  005300  177777                    .WORD   177777
 2913  005302  177777                    .WORD   177777
```

```
 2914  005304  167356                        .WORD   167356
 2915  005306  156735                        .WORD   156735
 2916  005310  146314                        .WORD   146314
 2917  005312  135673                        .WORD   135673
 2918  005314  125252                        .WORD   125252
 2919  005316  114631                        .WORD   114631
 2920  005320  104210                        .WORD   104210
 2921  005322  073567                        .WORD   073567
 2922  005324  063146                        .WORD   063146
 2923  005326  052525                        .WORD   052525
 2924  005330  042104                        .WORD   042104
 2925  005332  031463                        .WORD   031463
 2926  005334  021042                        .WORD   021042
 2927  005336  010421                        .WORD   010421
 2928  005340  000000                        .WORD   000000
 2929
 2930  005342  000000                PAT5:   .WORD   000000
 2931  005344  177777                        .WORD   177777
 2932  005346  000000                        .WORD   000000
 2933  005350  000000                        .WORD   000000
 2934  005352  177777                        .WORD   177777
 2935  005354  177777                        .WORD   177777
 2936  005356  000000                        .WORD   000000
 2937  005360  000000                        .WORD   000000
 2938  005362  000000                        .WORD   000000
 2939  005364  177777                        .WORD   177777
 2940  005366  177777                        .WORD   177777
 2941  005370  177777                        .WORD   177777
 2942  005372  000000                        .WORD   000000
 2943  005374  000000                        .WORD   000000
 2944  005376  000000                        .WORD   000000
 2945  005400  000000                        .WORD   000000
 2946  005402  177777                        .WORD   177777
 2947  005404  177777                        .WORD   177777
 2948  005406  177777                        .WORD   177777
 2949  005410  177777                        .WORD   177777
 2950  005412  000000                        .WORD   000000
 2951  005414  000000                        .WORD   000000
 2952  005416  000000                        .WORD   000000
 2953  005420  177777                        .WORD   177777
 2954  005422  177777                        .WORD   177777
 2955  005424  177777                        .WORD   177777
 2956  005426  000000                        .WORD   000000
 2957  005430  000000                        .WORD   000000
 2958  005432  177777                        .WORD   177777
 2959  005434  177777                        .WORD   177777
 2960  005436  000000                        .WORD   000000
 2961  005440  177777                        .WORD   177777
 2962
 2963  005442  000001                PAT6:   .WORD   000001
 2964  005444  000003                        .WORD   000003
 2965  005446  000007                        .WORD   000007
 2966  005450  000017                        .WORD   000017
 2967  005452  000037                        .WORD   000037
 2968  005454  000077                        .WORD   000077
 2969  005456  000177                        .WORD   000177
```

```
 2970  005460  000377                    .WORD   000377
 2971  005462  000777                    .WORD   000777
 2972  005464  001777                    .WORD   001777
 2973  005466  003777                    .WORD   003777
 2974  005470  007777                    .WORD   007777
 2975  005472  017777                    .WORD   017777
 2976  005474  037777                    .WORD   037777
 2977  005476  077777                    .WORD   077777
 2978  005500  177777                    .WORD   177777
 2979  005502  177776                    .WORD   177776
 2980  005504  177774                    .WORD   177774
 2981  005506  177770                    .WORD   177770
 2982  005510  177760                    .WORD   177760
 2983  005512  177740                    .WORD   177740
 2984  005514  177700                    .WORD   177700
 2985  005516  177600                    .WORD   177600
 2986  005520  177400                    .WORD   177400
 2987  005522  177000                    .WORD   177000
 2988  005524  176000                    .WORD   176000
 2989  005526  174000                    .WORD   174000
 2990  005530  170000                    .WORD   170000
 2991  005532  160000                    .WORD   160000
 2992  005534  140000                    .WORD   140000
 2993  005536  100000                    .WORD   100000
 2994  005540  000000                    .WORD   000000
 2995
 2996  005542  000001            PAT7:   .WORD   000001
 2997  005544  000002                    .WORD   000002
 2998  005546  000004                    .WORD   000004
 2999  005550  000010                    .WORD   000010
 3000  005552  000020                    .WORD   000020
 3001  005554  000040                    .WORD   000040
 3002  005556  000100                    .WORD   000100
 3003  005560  000200                    .WORD   000200
 3004  005562  000400                    .WORD   000400
 3005  005564  001000                    .WORD   001000
 3006  005566  002000                    .WORD   002000
 3007  005570  004000                    .WORD   004000
 3008  005572  010000                    .WORD   010000
 3009  005574  020000                    .WORD   020000
 3010  005576  040000                    .WORD   040000
 3011  005600  100000                    .WORD   100000
 3012  005602  100000                    .WORD   100000
 3013  005604  040000                    .WORD   040000
 3014  005606  020000                    .WORD   020000
 3015  005610  010000                    .WORD   010000
 3016  005612  004000                    .WORD   004000
 3017  005614  002000                    .WORD   002000
 3018  005616  001000                    .WORD   001000
 3019  005620  000400                    .WORD   000400
 3020  005622  000200                    .WORD   000200
 3021  005624  000100                    .WORD   000100
 3022  005626  000040                    .WORD   000040
 3023  005630  000020                    .WORD   000020
 3024  005632  000010                    .WORD   000010
 3025  005634  000004                    .WORD   000004
```

```
3026  005636  000002                    .WORD   000002
3027  005640  000001                    .WORD   000001
3028
3029  005642  177776          PAT10:    .WORD   177776
3030  005644  177775                    .WORD   177775
3031  005646  177773                    .WORD   177773
3032  005650  177767                    .WORD   177767
3033  005652  177757                    .WORD   177757
3034  005654  177737                    .WORD   177737
3035  005656  177677                    .WORD   177677
3036  005660  177577                    .WORD   177577
3037  005662  177377                    .WORD   177377
3038  005664  176777                    .WORD   176777
3039  005666  175777                    .WORD   175777
3040  005670  173777                    .WORD   173777
3041  005672  167777                    .WORD   167777
3042  005674  157777                    .WORD   157777
3043  005676  137777                    .WORD   137777
3044  005700  077777                    .WORD   077777
3045  005702  077777                    .WORD   077777
3046  005704  137777                    .WORD   137777
3047  005706  157777                    .WORD   157777
3048  005710  167777                    .WORD   167777
3049  005712  173777                    .WORD   173777
3050  005714  175777                    .WORD   175777
3051  005716  176777                    .WORD   176777
3052  005720  177377                    .WORD   177377
3053  005722  177577                    .WORD   177577
3054  005724  177677                    .WORD   177677
3055  005726  177737                    .WORD   177737
3056  005730  177757                    .WORD   177757
3057  005732  177767                    .WORD   177767
3058  005734  177773                    .WORD   177773
3059  005736  177775                    .WORD   177775
3060  005740  177776                    .WORD   177776
3061
3062  005742  172666          PAT11:    .WORD   172666
3063  005744  155555                    .WORD   155555
3064  005746  172666                    .WORD   172666
3065  005750  155555                    .WORD   155555
3066  005752  172666                    .WORD   172666
3067  005754  155555                    .WORD   155555
3068  005756  172666                    .WORD   172666
3069  005760  155555                    .WORD   155555
3070  005762  172666                    .WORD   172666
3071  005764  155555                    .WORD   155555
3072  005766  172666                    .WORD   172666
3073  005770  155555                    .WORD   155555
3074  005772  172666                    .WORD   172666
3075  005774  155555                    .WORD   155555
3076  005776  172666                    .WORD   172666
3077  006000  155555                    .WORD   155555
3078  006002  172666                    .WORD   172666
3079  006004  155555                    .WORD   155555
3080  006006  172666                    .WORD   172666
3081  006010  155555                    .WORD   155555
```

```
 3082  006012  172666                    .WORD   172666
 3083  006014  155555                    .WORD   155555
 3084  006016  172666                    .WORD   172666
 3085  006020  155555                    .WORD   155555
 3086  006022  172666                    .WORD   172666
 3087  006024  155555                    .WORD   155555
 3088  006026  172666                    .WORD   172666
 3089  006030  155555                    .WORD   155555
 3090  006032  172666                    .WORD   172666
 3091  006034  155555                    .WORD   155555
 3092  006036  172666                    .WORD   172666
 3093  006040  155555                    .WORD   155555
 3094
 3095  006042  153333            PAT12:  .WORD   153333
 3096  006044  066667                    .WORD   066667
 3097  006046  153333                    .WORD   153333
 3098  006050  066667                    .WORD   066667
 3099  006052  153333                    .WORD   153333
 3100  006054  066667                    .WORD   066667
 3101  006056  153333                    .WORD   153333
 3102  006060  066667                    .WORD   066667
 3103  006062  153333                    .WORD   153333
 3104  006064  066667                    .WORD   066667
 3105  006066  153333                    .WORD   153333
 3106  006070  066667                    .WORD   066667
 3107  006072  153333                    .WORD   153333
 3108  006074  066667                    .WORD   066667
 3109  006076  153333                    .WORD   153333
 3110  006100  066667                    .WORD   066667
 3111  006102  153333                    .WORD   153333
 3112  006104  066667                    .WORD   066667
 3113  006106  153333                    .WORD   153333
 3114  006110  066667                    .WORD   066667
 3115  006112  153333                    .WORD   153333
 3116  006114  066667                    .WORD   066667
 3117  006116  153333                    .WORD   153333
 3118  006120  066667                    .WORD   066667
 3119  006122  153333                    .WORD   153333
 3120  006124  066667                    .WORD   066667
 3121  006126  153333                    .WORD   153333
 3122  006130  066667                    .WORD   066667
 3123  006132  153333                    .WORD   153333
 3124  006134  066667                    .WORD   066667
 3125  006136  153333                    .WORD   153333
 3126  006140  066667                    .WORD   066667
 3127
 3128  006142  000000            PAT13:  .WORD   000000
 3129  006144  177777                    .WORD   177777
 3130  006146  177777                    .WORD   177777
 3131  006150  177777                    .WORD   177777
 3132  006152  177777                    .WORD   177777
 3133  006154  177777                    .WORD   177777
 3134  006156  177777                    .WORD   177777
 3135  006160  177777                    .WORD   177777
 3136  006162  177777                    .WORD   177777
 3137  006164  177777                    .WORD   177777
```

```
3138  006166  177777                    .WORD   177777
3139  006170  177777                    .WORD   177777
3140  006172  177777                    .WORD   177777
3141  006174  177777                    .WORD   177777
3142  006176  177777                    .WORD   177777
3143  006200  177777                    .WORD   177777
3144  006202  177777                    .WORD   177777
3145  006204  177777                    .WORD   177777
3146  006206  177777                    .WORD   177777
3147  006210  177777                    .WORD   177777
3148  006212  177777                    .WORD   177777
3149  006214  177777                    .WORD   177777
3150  006216  177777                    .WORD   177777
3151  006220  177777                    .WORD   177777
3152  006222  177777                    .WORD   177777
3153  006224  177777                    .WORD   177777
3154  006226  177777                    .WORD   177777
3155  006230  177777                    .WORD   177777
3156  006232  177777                    .WORD   177777
3157  006234  177777                    .WORD   177777
3158  006236  177777                    .WORD   177777
3159  006240  177777                    .WORD   177777
3160
3161  006242  177777            PAT14:  .WORD   177777
3162  006244  000000                    .WORD   000000
3163  006246  000000                    .WORD   000000
3164  006250  000000                    .WORD   000000
3165  006252  000000                    .WORD   000000
3166  006254  000000                    .WORD   000000
3167  006256  000000                    .WORD   000000
3168  006260  000000                    .WORD   000000
3169  006262  000000                    .WORD   000000
3170  006264  000000                    .WORD   000000
3171  006266  000000                    .WORD   000000
3172  006270  000000                    .WORD   000000
3173  006272  000000                    .WORD   000000
3174  006274  000000                    .WORD   000000
3175  006276  000000                    .WORD   000000
3176  006300  000000                    .WORD   000000
3177  006302  000000                    .WORD   000000
3178  006304  000000                    .WORD   000000
3179  006306  000000                    .WORD   000000
3180  006310  000000                    .WORD   000000
3181  006312  000000                    .WORD   000000
3182  006314  000000                    .WORD   000000
3183  006316  000000                    .WORD   000000
3184  006320  000000                    .WORD   000000
3185  006322  000000                    .WORD   000000
3186  006324  000000                    .WORD   000000
3187  006326  000000                    .WORD   000000
3188  006330  000000                    .WORD   000000
3189  006332  000000                    .WORD   000000
3190  006334  000000                    .WORD   000000
3191  006336  000000                    .WORD   000000
3192  006340  000000                    .WORD   000000
3193
```

```
3194  006342  172304                PAT15:  .WORD   172304
3195  006344  172304                        .WORD   172304
3196  006346  172304                        .WORD   172304
3197  006350  172304                        .WORD   172304
3198  006352  172304                        .WORD   172304
3199  006354  172304                        .WORD   172304
3200  006356  172304                        .WORD   172304
3201  006360  172304                        .WORD   172304
3202  006362  172304                        .WORD   172304
3203  006364  172304                        .WORD   172304
3204  006366  172304                        .WORD   172304
3205  006370  172304                        .WORD   172304
3206  006372  172304                        .WORD   172304
3207  006374  172304                        .WORD   172304
3208  006376  172304                        .WORD   172304
3209  006400  172304                        .WORD   172304
3210  006402  172304                        .WORD   172304
3211  006404  172304                        .WORD   172304
3212  006406  172304                        .WORD   172304
3213  006410  172304                        .WORD   172304
3214  006412  172304                        .WORD   172304
3215  006414  172304                        .WORD   172304
3216  006416  172304                        .WORD   172304
3217  006420  172304                        .WORD   172304
3218  006422  172304                        .WORD   172304
3219  006424  172304                        .WORD   172304
3220  006426  172304                        .WORD   172304
3221  006430  172304                        .WORD   172304
3222  006432  172304                        .WORD   172304
3223  006434  172304                        .WORD   172304
3224  006436  172304                        .WORD   172304
3225  006440  172304                        .WORD   172304
3226
3227  006442  070627                PAT16:  .WORD   070627
3228  006444  113431                        .WORD   113431
3229  006446  014561                        .WORD   014561
3230  006450  070627                        .WORD   070627
3231  006452  113431                        .WORD   113431
3232  006454  014561                        .WORD   014561
3233  006456  070627                        .WORD   070627
3234  006460  113431                        .WORD   113431
3235  006462  014561                        .WORD   014561
3236  006464  070627                        .WORD   070627
3237  006466  113431                        .WORD   113431
3238  006470  014561                        .WORD   014561
3239  006472  070627                        .WORD   070627
3240  006474  113431                        .WORD   113431
3241  006476  014561                        .WORD   014561
3242  006500  070627                        .WORD   070627
3243  006502  113431                        .WORD   113431
3244  006504  014561                        .WORD   014561
3245  006506  070627                        .WORD   070627
3246  006510  113431                        .WORD   113431
3247  006512  014561                        .WORD   014561
3248  006514  070627                        .WORD   070627
3249  006516  113431                        .WORD   113431
```

```
 3250  006520  014561                        .WORD   014561
 3251  006522  070627                        .WORD   070627
 3252  006524  113431                        .WORD   113431
 3253  006526  014561                        .WORD   014561
 3254  006530  070627                        .WORD   070627
 3255  006532  113431                        .WORD   113431
 3256  006534  014561                        .WORD   014561
 3257  006536  070627                        .WORD   070627
 3258  006540  113431                        .WORD   113431
 3259
 3260  006542  133467              PAT17:    .WORD   133467
```

```
 3261  006544  133467                    .WORD   133467
 3262  006546  133467                    .WORD   133467
 3263  006550  133467                    .WORD   133467
 3264  006552  133467                    .WORD   133467
 3265  006554  133467                    .WORD   133467
 3266  006556  133467                    .WORD   133467
 3267  006560  133467                    .WORD   133467
 3268  006562  133467                    .WORD   133467
 3269  006564  133467                    .WORD   133467
 3270  006566  133467                    .WORD   133467
 3271  006570  133467                    .WORD   133467
 3272  006572  133467                    .WORD   133467
 3273  006574  133467                    .WORD   133467
 3274  006576  133467                    .WORD   133467
 3275  006600  133467                    .WORD   133467
 3276  006602  133467                    .WORD   133467
 3277  006604  133467                    .WORD   133467
 3278  006606  133467                    .WORD   133467
 3279  006610  133467                    .WORD   133467
 3280  006612  133467                    .WORD   133467
 3281  006614  133467                    .WORD   133467
 3282  006616  133467                    .WORD   133467
 3283  006620  133467                    .WORD   133467
 3284  006622  133467                    .WORD   133467
 3285  006624  133467                    .WORD   133467
 3286  006626  133467                    .WORD   133467
 3287  006630  133467                    .WORD   133467
 3288  006632  133467                    .WORD   133467
 3289  006634  133467                    .WORD   133467
 3290  006636  133467                    .WORD   133467
 3291  006640  133467                    .WORD   133467
```

```
                                        .SBTTL  PROGRAM SETUP
                                        
006642  112737  000001  001361  PARSPT: MOVB    #1,FLAG         ;SET FLAG FOR PARAMETER ENTRY
006650  000412                          BR      PRGSRT          ;START PROGRAM
                                        
006652  112737  177777  001361  RESTRT: MOVB    #-1,FLAG        ;SET FLAG FOR RETRY
006660  000406                          BR      PRGSRT          ;START PROGRAM
                                        
006662  105037  001361          START1: CLRB    FLAG            ;SET FOR ENTRY INPUT START
006666  000403                          BR      PRGSRT          ;START PROGRAM
                                        
006670  112737  000002  001361  START:  MOVB    #2,FLAG         ;SET FOR AUTO SIZE PGM SIZE
                                        
006676  012737  000340  177776  PRGSRT: MOV     #PR7,PS         ;LOCK OUT ALL INTERRUPTS
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
006704  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
006710  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
006712  022706  001140                  CMP     #SWR,R6 ;;DONE?
006716  001374                          BNE     .-6             ;;LOOP BACK IF NO
006720  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
006724  012737  056160  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
006732  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
006740  012737  055046  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
006746  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
006754  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
006760  012737  007014  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
006766  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
006774  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
007002  022777  177777  172130          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
007010  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
007012  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
007014  012716  007022          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
007020  000002                          RTI
007022  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
007030  012737  000174  001142          MOV     #DISPREG,DISPLAY
007036  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
                                        
007042  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
007046  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
007054  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
007056  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
007064                          67$:
                                        .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
007064  005227  177777                  INC     #-1             ;;FIRST TIME?
007070  001040                          BNE     68$             ;;BRANCH IF NO
007072  104401  007100                  TYPE    ,69$            ;;TYPE ASCIZ STRING
007076  000435                          BR      68$             ;;GET OVER THE ASCIZ
007172                          ;;69$:  .ASCIZ  <CRLF>'RK611/RK06-RK07 PERFORMANCE EXERCISER: MAINDEC CZR6P-D'<CRLF>
                                68$:
007172  012737  000000  000036          MOV     #PR0,TRAPVEC+2  ;SET TRAP VECTOR TO PRIORITY 0
```

```
3348  007200  000005                  PWRSRT: RESET                   ;RESET SYSTEM
3349  007202  004737  054714                  JSR     PC,$SIZE        ;GET MAXIMUM MEMORY ADDRESS
3350  007206  012737  007260  000004          MOV     #20$,ERRVEC     ;SET VECTOR FOR MEMORY PARITY CHECK
3351  007214  012737  000340  000006          MOV     #PR7,ERRVEC+2
3352  007222  012703  172100                  MOV     #MEMBAS,R3      ;LOAD REGISTER TO DTERMIN IF
3353                                                                  ;  MEMORY CHECK ENABLE ABAILIABLE
3354  007226  012704  000014                  MOV     #12.,R4         ;LOAD COUNT
3355  007232  012723  000001          16$:    MOV     #PAR.EN,(R3)+   ;ENABLE MEMORY CHECK VECTORY
3356  007236  012737  052650  000114          MOV     #MEMERR,MEMVEC  ;LOAD MEMORY CHECK VECTOR
3357  007244  012737  000340  000116          MOV     #PR7,MEMVEC+2
3358  007252  005304                          DEC     R4              ;CHECK IF FINISHED
3359  007254  001366                          BNE     16$             ;NO, SET UP NEXT MEMORY PARITY MODULE
3360  007256  000401                          BR      22$             ;RESTORE TRAP VECTOR
3361
3362  007260  022626                  20$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
3363  007262  012737  000006  000004  22$:    MOV     #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
3364  007270  005037  000006                  CLR     ERRVEC+2
3365  007274  012737  000001  001670  START$: MOV     #1,FBLKC        ;INITIALIZE FREE BLOCK COUNT
3366  007302  012703  001672                  MOV     #FBLKT,R3       ;STORE FREE BLOCK TABLE ADDRESS
3367  007306  012704  000022                  MOV     #18.,R4         ;LOAD COUNT FOR FREE BLOCK CLEAR
3368  007312  005023                  2$:     CLR     (R3)+           ;CLEAR FREE BLOCK TABLE
3369  007314  005304                          DEC     R4              ;DECREMENT COUNT
3370  007316  001375                          BNE     2$              ;CHECK IF FINISHED
3371  007320  012737  066610  001672          MOV     #BUFADD,FBLKT   ;LOAD FIRST ENTRY IN FREE BLOCK TABLE
3372  007326  105037  001644                  CLRB    RTYSCN          ;CLEAR RETRY SECTOR COUNT
3373  007332  005037  001566                  CLR     ERRNO           ;CLEAR ADDRESS OF PARAMETER BLOCK
3374                                                                  ;  PROCESSING ERROR
3375  007336  005037  001462                  CLR     E.CONT          ;CLEAR CONTROLLER ERROR FLAGS
3376  007342  005037  001564                  CLR     ERCONT          ;CLEAR RECOVERABLE CONTROLLER ERROR FLAGS
3377  007346  005037  001626                  CLR     DLTCNT          ;CLEAR DATA LATE COUNT
3378  007352  005037  001630                  CLR     CNTCNT          ;CLEAR OTHER ERRORS COUNT
3379  007356  105037  001633                  CLRB    STATIS          ;CLEAR INTERVAL STATISTICS FLAG
3380  007362  112737  177777  001513          MOVB    #-1,O.OVER      ;SET ALL DRIVES FOR IMPLIED SEEKS
3381  007370  012703  001314                  MOV     #AVAILQ,R3      ;STORE QUEUE BASE ADDRESS
3382  007374  012704  000010                  MOV     #8.,R4          ;LOAD QUEUE COUNT
3383  007400  005023                  3$:     CLR     (R3)+           ;CLEAR QUEUE HEADS AND TAILS
3384  007402  005304                          DEC     R4              ;DECREMENT COUNT
3385  007404  001375                          BNE     3$              ;CHECK IF FINISHED
3386  007406  012704  001544                  MOV     #W.DRV,R4       ;LOAD ADDRESS OF TIME OUT COUNTS
3387  007412  012703  000010                  MOV     #8.,R3          ;LOAD COUNT
3388  007416  005024                  4$:     CLR     (R4)+           ;CLEAR DRIVE TIME OUT COUNTS
3389  007420  005303                          DEC     R3              ;DECREMENT COUNT
3390  007422  001375                          BNE     4$              ;CHECK IF FINISHED
3391  007424  004737  043550                  JSR     PC,CHKCLK       ;CHECK IF CLOCK IS PRESENT
3392  007430  105037  001512                  CLRB    W.TIME          ;CLEAR DRIVES BEING TIMED
3393  007434  013737  001470  001466          MOV     W.MILI,W.MTIM   ;INITIALIZE MILI-SECOND TIMER
3394  007442  005037  001464                  CLR     O.WAIT          ;CLEAR PARAMETER BLOCK FOR DRIVE
3395                                                                  ;  WAITING COMMAND COMPLETION
3396  007446  105037  001506                  CLRB    I.ISRL          ;RESET RELEASE OR CONTROLLER CLEAR ISSUED
3397  007452  105737  001361                  TSTB    FLAG            ;CHECK IF RESTART
3398  007456  100072                          BPL     RESTO           ;NO, CONTINUE
3399
3400  007460  013737  055044  001674          MOV     $LSTAD,FBLKT+2  ;LOAD MAX. ADD IF FREE BLOCK TABLE
3401  007466  105737  001365                  TSTB    OVLYLD          ;CHECK IF OVERLAY LOADER
3402  007472  001012                          BNE     20$             ;YES, LOAD RK06 VECTOR ADDRESS
3403  007474  105737  000041                  TSTB    41              ;SEE WHO LOADED THE PROBLEM
```

```
3404  007500  001004                        BNE     10$                 ;BRANCH IF XXDP
3405  007502  162737  000300  001674        SUB     #96.*2.,FBLKT+2     ;SUBTRACT ABS LOADER SIZE
3406  007510  000403                        BR      20$                 ;LOAD VECTOR ADDRESS
3407
3408  007512  162737  004004  001674  10$:  SUB     #1026.*2.,FBLKT+2   ;SUBTRACT XXDP LOADER
3409  007520  013704  001450          20$:  MOV     RKVEC,R4            ;STORE VECTOR ADDRESS
3410  007524  012724  044444                MOV     #I.INTR,(R4)+       ;LOAD RK06 VECTOR ADDRESS
3411  007530  013714  001452                MOV     RKPRI,(R4)          ;  AND INTERRUPT PRIORITY
3412  007534  012703  001524                MOV     #PBLKT,R3           ;LOAD PARAMETER BLOCK TABLE ADDRESS
3413  007540  012704  000011                MOV     #9.,R4              ;LOAD COUNT
3414
3415  007544  005304                  21$:  DEC     R4                  ;DECREMENT COUNT AND CHECK IF DONE
3416  007546  001434                        BEQ     30$                 ;YES, SET UP TTY INTERRUPT
3417  007550  012305                        MOV     (R3)+,R5            ;GET PARAMETER BLOCK ADDRESS
3418  007552  105065  000145                CLRB    P.BUFF(R5)          ;CLEAR BUFFER ALLOCATED
3419  007556  005065  000212                CLR     P.ERR(R5)           ;CLEAR ERROR FLAGS AND RETRY COUNTS
3420  007562  005065  000150                CLR     P.DSTT(R5)
3421  007566  005065  000146                CLR     P.RECT(R5)
3422
3423
3424                                  ;       RESET THE FOLLOWING DRIVER FLAGS
3425                                  ;               DRIVE POSITIONING
3426                                  ;               DRIVE POSITIONING FOR DATA TRANSFER
3427                                  ;               DRIVE SEIZED
3428                                  ;               DRIVE UNLOADED DUE TO ERROR
3429                                  ;               PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
3430  007572  042765  070006  000014        BIC     #DRVPOS!DRVPDT!DRVSZD!E.UNLD!Q.INIT,P.PRST(R5)
3431
3432  007600  032765  000001  000014        BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
3433  007606  001756                        BEQ     21$                 ;NO, GET NEXT PARAMETER BLOCK
3434  007610  105765  000210                TSTB    P.ASSN(R5)          ;CHECK IF DRIVE IS BEING ASSIGNED
3435  007614  001003                        BNE     22$                 ;YES, RESTART ASSIGNMENT SEQUENCE
3436  007616  004737  013612                JSR     PC,TEST1            ;ISSUE A START SPINDLE IF NECESSARY
3437  007622  000750                        BR      21$                 ;GET NEXT PARAMETER BLOCK
3438
3439  007624  142765  177757  000210  22$:  BICB    #^C<BIT4>,P.ASSN(R5) ;KEEP WRITE PACK BIT
3440  007632  004737  013536                JSR     PC,TEST             ;INITIATE DRIVE ASSIGNMENT SEQUENCE
3441  007636  000742                        BR      21$                 ;GET NEXT PARAMETER BLOCK
3442
3443  007640  000137  011260          30$:  JMP     REST2               ;SET TTY INTERRUPT
3444
3445  007644  012737  176543  055422  RESTO:  MOV   #176543,$HINUM      ;INITIALIZE RANDOM NUMBER GENERATOR
3446  007652  012737  123456  055424        MOV     #123456,$LONUM
3447  007660  005037  001372                CLR     SECOND              ;INITIALIZE INTERNAL CLOCK
3448  007664  005037  001370                CLR     MINUTE
3449  007670  005037  001366                CLR     HOUR
3450  007674  012703  001524                MOV     #PBLKT,R3           ;LOAD PARAMETER BLOCK TABLE ADDRESS
3451  007700  012704  000010                MOV     #8.,R4              ;LOAD COUNT
3452  007704  012305                  4$:   MOV     (R3)+,R5            ;GET PARAMETER BLOCK ADDRESS
3453  007706  005065  000014                CLR     P.PRST(R5)          ;CLEAR RK06 PROGRAM DEVICE STATUS REG.
3454  007712  005065  000212                CLR     P.ERR(R5)           ;CLEAR DRIVE ERROR FLAGS
3455  007716  105065  000210                CLRB    P.ASSN(R5)          ;CLEAR DRIVE ASSIGNMENT CODE
3456  007722  010500                        MOV     R5,R0               ;STORE R0 FOR TRACK EXCLUSIONS
3457  007724  062700  000232                ADD     #P.EXAR,R0          ;DETERMINE TRACK EXCLUSION BLOCK
3458  007730  012701  000012                MOV     #10.,R1             ;LOAD COUNT FOR CLEARING TRACK EXCLUSION
3459                                                                    ;  BLOCK
```

```
3460  007734  005020                  5$:     CLR     (R0)+           ;CLEAR TRACK EXCLUSION ENTRY
3461  007736  005301                          DEC     R1              ;DECREMENT COUNT
3462  007740  001375                          BNE     5$              ;CHECK IF WHOLE BLOCK CLEARED
3463  007742  005304                          DEC     R4              ;DECREMENT COUNT
3464  007744  001357                          BNE     4$              ;TEST IF DONE
3465  007746  122737  000001  001361          CMPB    #1,FLAG         ;CHECK IF PARAMETER START
3466  007754  001402                          BEQ     10$             ;YES, ASK FOR PARAMETERS
3467  007756  000137  011110                  JMP     S1$             ;USE DEFAULT VALUES
3468
3469  007762  104400  ,7557          10$:    TYPE    ,SYS000         ;TYPE "CPU ID"
3470  007766  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3471  007772  010022                          11$                     ;COMMA DETECTED
3472  007774  010040                          13$                     ;CARRIAGE RETURN DETECTED
3473  007776  011110                          S1$                     ;CONTROL Z <^Z> DETECTED
3474  010000  007762                          10$                     ;CONTROL C <^C> DETECTED
3475  010002  010346                          MOV     R3,-(SP)        ;LOAD BUFFER ADDRESS ON STACK
3476  010004  004737  054406                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
3477  010010  010022                          11$                     ;ERROR RETURN
3478  010012  012604                          MOV     (SP)+,R4        ;STORE INPUT VALUE
3479  010014  022704  000400                  CMP     #400,R4         ;CHECK IF 0-377
3480  010020  101010                          BHI     14$             ;YES, LOAD CPU ID
3481  010022                          11$:
3482  010022  010337  010030                  MOV     R3,64$          ;LOAD BUFFER ADDRESS
3483  010026  104401                          TYPE                    ;TYPE RECEIVED INPUT
3484  010030  000000                  64$:    .WORD   0               ;BUFFER ADDRESS
3485  010032  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
3486  010036  000751                          BR      10$             ;TRY AGAIN
3487
3488  010040  005004                  13$:    CLR     R4              ;CLEAR CPU ID (R4)
3489  010042  110437  001364          14$:    MOVB    R4,CPUID        ;LOAD CPU ID
3490
3491  010046  013737  055044  001674  15$:    MOV     $LSTAD,FBLKT+2  ;LOAD LAST ADDRESS IN FIRST
3492                                                                  ;  FREE BLOCK ENTRY
3493  010054  104401  057567                  TYPE    ,SYS001         ;TYPE "OVERLAY LOADER"
3494  010060  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3495  010064  010134                          18$                     ;COMMA DETECTED
3496  010066  010102                          16$                     ;CARRIAGE RETURN DETECTED
3497  010070  011122                          S2$                     ;CONTROL Z <^Z> DETECTED
3498  010072  007762                          10$                     ;CONTROL C <^C> DETECTED
3499  010074  122713  000131                  CMPB    #'Y,(R3)        ;CHECK IF YES
3500  010100  001424                          BEQ     19$             ;YES, INDICATE THAT LOADER HAS BEEN OVERLAYED
3501  010102  105037  001365          16$:    CLRB    OVLYLD          ;INDICATE THAT LOADER HAS NOT BEEN OVERLAYED
3502  010106  105737  000041                  TSTB    41              ;SEE WHO LOADED THE PROGRAM
3503  010112  001004                          BNE     17$             ;BRANCH IF XXDP
3504  010114  162737  000300  001674          SUB     #96.*2.,FBLKT+2 ;SUBTRACT ABS LOADER SIZE
3505  010122  000416                          BR      20$             ;ASK FOR MAXIMUM TRANSFER SIZE
3506
3507  010124  162737  006000  001674  17$:    SUB     #1536.*2.,FBLKT+2 ;SUBTRACT XXDP LOADER SIZE
3508  010132  000412                          BR      20$             ;ASK FOR TRANSFER SIZE
3509
3510  010134                          18$:
3511  010134  010337  010142                  MOV     R3,65$          ;LOAD BUFFER ADDRESS
3512  010140  104401                          TYPE                    ;TYPE RECEIVED INPUT
3513  010142  000000                  65$:    .WORD   0               ;BUFFER ADDRESS
3514  010144  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
3515  010150  000736                          BR      15$             ;TRY AGAIN
```

```
3516
3517  010152  112737  177777  001365  19$:    MOVB    #-1,OVLYLD      ;INDICATE THAT LOADER HAS BEEN OVERLAYED
3518
3519  010160  104401  057607          20$:    TYPE    ,SYS002         ;TYPE 'MAXIMUM TRANSFER''
3520  010164  013746  001674                  MOV     FBLKT+2,-(SP)   ;STORE LAST USABLE MEMORY LOCATION
3521  010170  163716  001672                  SUB     FBLKT,(SP)      ;CALCULATE MEMORY LEFT
3522  010174  022716  027000                  CMP     #256.*23.*2.,(SP) ;CHECK IF GREATER THAN OF EQUAL TO 5888
3523  010200  101402                          BLOS    21$             ;YES, USE 5888
3524  010202  006216                          ASR     (SP)            ;DETERMINE NUMBER OF WORDS
3525  010204  000402                          BR      22$             ;TYPE VALUE
3526
3527  010206  012716  013400          21$:    MOV     #256.*23.,(SP)  ;LOAD 5888
3528
3529  010212  012637  001570          22$:    MOV     (SP)+,TMPCNT    ;SAVE TRANSFER COUNT
3530  010216  013746  001570                  MOV     TMPCNT,-(SP)    ;SAVE TRANSFER COUNT FOR TYPE OUT
3531  010222  004737  053224                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
3532  010226  104401  057641                  TYPE    ,SYS003
3533  010232  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3534  010236  010300                          24$                     ;COMMA DETECTED
3535  010240  010316                          26$                     ;CARRIAGE RETURN DETECTED
3536  010242  011152                          S3$                     ;CONTROL Z <^Z> DETECTED
3537  010244  007762                          10$                     ;CONTROL C <^C> DETECTED
3538  010246  010346                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS ON STACK
3539  010250  004737  054406                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
3540  010254  010276                          23$                     ;ERROR RETURN
3541  010256  012604                          MOV     (SP)+,R4        ;STORE INPUT
3542  010260  001407                          BEQ     24$             ;CHECK IF ZERO
3543  010262  020437  001570                  CMP     R4,TMPCNT       ;CHECK IF LEGAL ENTRY
3544  010266  101004                          BHI     24$             ;NO, TRY AGAIN
3545  010270  010437  001354                  MOV     R4,MAXBUF       ;LOAD MAXIMUM TRANSFER
3546  010274  000422                          BR      27$             ;GET    NO. OF SOFT COMPARES
3547
3548  010276  005726                  23$:    TST     (SP)+           ;ADJUST STACK
3549  010300                          24$:
3550  010300  010337  010306                  MOV     R3,66$          ;LOAD BUFFER ADDRESS
3551  010304  104401                          TYPE                    ;TYPE RECEIVED INPUT
3552  010306  000000                  66$:    .WORD   0               ;BUFFER ADDRESS
3553  010310  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
3554  010314  000721                          BR      20$             ;TRY AGAIN
3555
3556  010316  013737  001570  001354  26$:    MOV     TMPCNT,MAXBUF ;LOAD MAXIMUM TRANSFER
3557  010324  022737  013400  001354          CMP     #256.*23.,MAXBUF ;CHECK IF LESS THAN OR EQUAL TO 23 SECTORS
3558  010332  103003                          BHIS    27$             ;YES, USE CALCULATED VALUE
3559  010334  012737  013400  001354          MOV     #256.*23.,MAXBUF ;USE 23 SECTORS
3560
3561  010342  104401  057645          27$:    TYPE    ,SYS004         ;TYPE 'NO. OF SOFTWARE COMPARISONS''
3562  010346  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3563  010352  010410                          29$                     ;COMMA DETECTED
3564  010354  010426                          30$                     ;CARRIAGE RETURN DETECTED
3565  010356  011210                          S4$                     ;CONTROL Z <^Z> DETECTED
3566  010360  007762                          10$                     ;CONTROL C <^C> DETECTED
3567  010362  010346                          MOV     R3,-(SP)        ;STORE PARMETER
3568  010364  004737  054542                  JSR     PC,DECBIN       ;CONVERT TO BINARY
3569  010370  010410                          29$                     ;ERROR RETURN
3570  010372  023716  001354                  CMP     MAXBUF,(SP)     ;CHECK IF LEGAL INPUT
3571  010376  103403                          BLO     28$             ;TYPE VALUE AND RETRY
```

```
3572  010400  012637  001362                  MOV     (SP)+,SOFCMP    ;LOAD NUMBER OF SOFTWARE COMPARES
3573  010404  000413                          BR      31$             ;DETERMINE RK06 UNIBUS ADDRESS
3574
3575  010406  005726                  28$:    TST     (SP)+           ;ADJUST STACK
3576  010410                          29$:
3577  010410  010337  010416                  MOV     R3,67$          ;LOAD BUFFER ADDRESS
3578  010414  104401                          TYPE                    ;TYPE RECEIVED INPUT
3579  010416  000000                  67$:    .WORD   0               ;BUFFER ADDRESS
3580  010420  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
3581  010424  000746                          BR      27$             ;TRY AGAIN
3582
3583  010426  012737  000003  001362  30$:    MOV     #3,SOFCMP       ;LOAD DEFAULT VALUE
3584
3585  010434  105737  001374          31$:    TSTB    CLKFLG          ;CHECK IF CLOCK ON SYSTEM
3586  010440  001434                          BEQ     41$             ;NO, CONTINUE
3587  010442  104401  060040                  TYPE    ,SYS009         ;TYPE "STATISTIC INTERVAL="
3588  010446  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3589  010452  010510                          34$                     ;COMMA DETECTED
3590  010454  010526                          35$                     ;CARRIAGE RETURN DETECTED
3591  010456  011216                          S5$                     ;CONTROL Z <^Z> DETECTED
3592  010460  007762                          10$                     ;CONTROL C <^C> DETECTED
3593  010462  010346                          MOV     R3,-(SP)        ;STORE NUMBER FOR CONVERSION
3594  010464  004737  054542                  JSR     PC,DECBIN       ;CONVERT TO BINARY
3595  010470  010510                          34$                     ;ILLEGAL NUMBER
3596  010472  012604                          MOV     (SP)+,R4        ;STORE NUMBER
3597  010474  022704  000377                  CMP     #255.,R4        ;CHECK IF GREATER THAN 255
3598  010500  103403                          BLO     34$             ;NO, TRY AGAIN
3599  010502  110437  001360                  MOVB    R4,PERINV       ;LOAD PERFORMANCE INTERVAL
3600  010506  000411                          BR      41$             ;CONTINUE
3601
3602  010510                          34$:
3603  010510  010337  010516                  MOV     R3,68$          ;LOAD BUFFER ADDRESS
3604  010514  104401                          TYPE                    ;TYPE RECEIVED INPUT
3605  010516  000000                  68$:    .WORD   0               ;BUFFER ADDRESS
3606  010520  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
3607  010524  000743                          BR      31$             ;TRY AGAIN
3608
3609  010526  105037  001360          35$:    CLRB    PERINV          ;SET UP FOR NO INTERVAL STATISTICS
3610
3611  010532  104401  057677          41$:    TYPE    ,SYS005         ;TYPE 'RK06 BUS ADD.'
3612  010536  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3613  010542  010656                          44$                     ;COMMA DETECTED
3614  010544  010674                          45$                     ;CARRIAGE RETURN DETECTED
3615  010546  011222                          S6$                     ;CONTROL Z <^Z> DETECTED
3616  010550  007762                          10$                     ;CONTROL C <^C> DETECTED
3617  010552  010346                          MOV     R3,-(SP)        ;STORE PARAMETER
3618  010554  004737  054406                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
3619  010560  010656                          44$                     ;ERROR RETURN
3620  010562  012604                          MOV     (SP)+,R4        ;STORE INPUT
3621  010564  022704  160000                  CMP     #160000,R4      ;CHECK IF I/O PAGE
3622  010570  101032                          BHI     44$             ;NO,TRY AGAIN
3623  010572  012737  010632  000004          MOV     #42$,ERRVEC     ;LOAD ERRVEC TO NON-EXISTENT MEMORY
3624  010600  013737  000340  000006          MOV     PR7,ERRVEC+2    ;  PRIORITY 7
3625  010606  005714                          TST     (R4)            ;CHECK FOR NON-EXISTENT MEMORY
3626  010610  012737  000000  000006          MOV     #<HALT>,ERRVEC+2 ;REINSTATE TRAP CATCHER
3627  010616  012737  000006  000004          MOV     #ERRVEC+2,ERRVEC
```

```
3628  010624  010437  001446                 MOV     R4,RKBAS        ;LEGAL I/O ADDRESS, LOAD RK06 BASE
3629  010630  000424                         BR      48$             ;GET VECTOR ADDRESS
3630
3631  010632  012737  000000  000006  42$:   MOV     #<HALT>,ERRVEC+2 ;REINSTATE TRAP CATCHER
3632  010640  012737  000006  000004         MOV     #ERRVEC+2,ERRVEC
3633  010646  062706  000004                 ADD     #4,SP           ;ADJUST STACK
3634  010652  104401  057745                 TYPE    ,SYS007         ;TYPE 'NON-EXISTENT MEMORY'
3635  010656                         44$:
3636  010656  010337  010664                 MOV     R3,69$          ;LOAD BUFFER ADDRESS
3637  010662  104401                         TYPE                    ;TYPE RECEIVED INPUT
3638  010664  000000                 69$:    .WORD   0               ;BUFFER ADDRESS
3639  010666  104401  001164                 TYPE    ,$QUES          ;TYPE QUESTION MARK
3640  010672  000717                         BR      41$             ;TRY AGAIN
3641
3642  010674  012737  177440  001446  45$:   MOV     #177440,RKBAS   ;LOAD DEFAULT BUS ADDRESS
3643
3644  010702  104401  057722          48$:   TYPE    ,SYS006         ;TYPE 'RK06 VEC ADD'
3645  010706  004037  020516                 JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3646  010712  010750                         50$                     ;COMMA DETECTED
3647  010714  010766                         51$                     ;CARRIAGE RETURN DETECTED
3648  010716  011230                         57$                     ;CONTROL Z <^Z> DETECTED
3649  010720  007762                         10$                     ;CONTROL C <^C> DETECTED
3650  010722  010346                         MOV     R3,-(SP)        ;STORE PARAMETER
3651  010724  004737  054406                 JSR     PC,OCTBIN       ;CONVERT TO BINARY
3652  010730  010750                         50$                     ;ERROR RETURN
3653  010732  022716  001000                 CMP     #1000,(SP)      ;CHECK IF VECTOR SPACE
3654  010736  101403                         BLOS    49$             ;NO, TRY AGAIN
3655  010740  012637  001450                 MOV     (SP)+,RKVEC     ;LOAD VECTOR ADDRESS
3656  010744  000413                         BR      53$             ;GET PRIORITY
3657
3658  010746  005726                 49$:    TST     (SP)+           ;ADJUST STACK
3659  010750                         50$:
3660  010750  010337  010756                 MOV     R3,70$          ;LOAD BUFFER ADDRESS
3661  010754  104401                         TYPE                    ;TYPE RECEIVED INPUT
3662  010756  000000                 70$:    .WORD   0               ;BUFFER ADDRESS
3663  010760  104401  001164                 TYPE    ,$QUES          ;TYPE QUESTION MARK
3664  010764  000746                         BR      48$             ;TRY AGAIN
3665
3666  010766  012737  000210  001450  51$:   MOV     #210,RKVEC      ;LOAD DEFAULT RK06 VECTOR
3667  010774  104401  060064          53$:   TYPE    ,SYS010         ;TYPE 'RK06 PRIOR='
3668  011000  004037  020516                 JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3669  011004  011062                         56$                     ;COMMA DETECTED
3670  011006  011100                         57$                     ;CARRIAGE RETURN DETECTED
3671  011010  011236                         58$                     ;CONTROL Z <^Z> DETECTED
3672  011012  007762                         10$                     ;CONTROL C <^C> DETECTED
3673  011014  010346                         MOV     R3,-(SP)        ;STORE PARAMETER
3674  011016  004737  054406                 JSR     PC,OCTBIN       ;CONVERT TO BINARY
3675  011022  011062                         56$                     ;ERROR RETURN
3676  011024  022716  000007                 CMP     #7,(SP)         ;CHECK IF OCTAL
3677  011030  103413                         BLO     55$
3678  011032  022716  000004                 CMP     #4,(SP)
3679  011036  101010                         BHI     55$
3680  011040  006316                         ASL     (SP)            ;SHIFT 5 BITS LEFT
3681  011042  006316                         ASL     (SP)
3682  011044  006316                         ASL     (SP)
3683  011046  006316                         ASL     (SP)
```

```
3684  011050  006316                        ASL     (SP)
3685  011052  012637  001452                MOV     (SP)+,RKPRI     ;LOAD PRIORITY
3686  011056  000472                        BR      RES             ;LOAD RK06 VECTOR AND TTY VECTOR
3687
3688  011060  005726                55$:    TST     (SP)+           ;ADJUST STACK
3689  011062                        56$:
3690  011062  010337  011070              MOV     R3,71$          ;LOAD BUFFER ADDRESS
3691  011066  104401                      TYPE                    ;TYPE RECEIVED INPUT
3692  011070  000000                71$:   .WORD   0               ;BUFFER ADDRESS
3693  011072  104401  001164              TYPE    ,$QUES          ;TYPE QUESTION MARK
3694  011076  000736                      BR      53$             ;TRY AGAIN
3695
3696  011100  012737  000240  001452 57$:   MOV     #PR5,RKPRI      ;LOAD PRIORITY
3697  011106  000456                      BR      REST1           ;LOAD RK06 VECTOR AND TTY VECTOR
3698
3699  011110  105037  001364          S1$:   CLRB    CPUID           ;LOAD DEFAULT CPU ID
3700  011114  013737  055044  001674        MOV     $LSTAD,FBLKT+2  ;LOAD LAST ADDRESS
3701  011122  105037  001365          S2$:   CLRB    OVLYLD          ;INDICATE THAT THE LOADER IS NOT OVERLAYED
3702  011126  105737  000041                TSTB    41              ;SEE WHO LOADED PROGRAM
3703  011132  001004                        BNF     1$              ;BRANCH IF XXDP
3704  011134  162737  000300  001674        SUB     #96.*2.,FBLKT+2 ;SUBTRACT ABS LOADER SIZE
3705  011142  000403                        BR      S3$             ;GO DETERMINE MAX BUFFER SIZE
3706
3707  011144  162737  006000  001674  1$:   SUB     #1536.*2.,FBLKT+2 ;SUBTRACT XXDP LOADER SIZE
3708  011152  013746  001674          S3$:   MOV     FBLKT+2,-(SP)   ;STORE MAXIMUM ADDRESS
3709  011156  163716  001672                SUB     FBLKT,(SP)      ;DETERMINE MEMORY AVAILIABLE
3710  011162  022716  027000                CMP     #256.*23.*2.,(SP) ;CHECK IF ENOUGH ROOM
3711  011166  101005                        BHI     1$              ;NO,LOAD MEMORY AVILIABLE
3712  011170  005726                        TST     (SP)+           ;ADJUST STACK
3713  011172  012737  013400  001354        MOV     #256.*23.,MAXBUF ;LOAD MAXIMUM BUFFER SIZE
3714  011200  000403                        BR      S4$             ;LOAD NUMBER OF SOFTWARE COMPARES
3715
3716  011202  006216                1$:     ASR     (SP)            ;CHANGE TO WORDS
3717  011204  012637  001354              MOV     (SP)+,MAXBUF    ;LOAD MAXIMUM WORDS TRANSFERRED
3718  011210  012737  000003  001362 S4$:   MOV     #3,SOFCMP       ;LOAD NO. OF SOFTWARE COMPARES
3719  011216  105037  001360          S5$:   CLRB    PERINV          ;SET UP FOR NO INTERVAL STATISTICS
3720  011222  012737  177440  001446 S6$:   MOV     #177440,RKBAS   ;LOAD DEFAULT BUS ADDRESS
3721  011230  012737  000210  001450 S7$:   MOV     #210,RKVEC      ;LOAD DEFAULT VECTOR ADDRESS
3722  011236  012737  000240  001452 S8$:   MOV     #PR5,RKPRI      ;LOAD RK06 PRIORITY
3723
3724  011244  013704  001450          REST1: MOV     RKVEC,R4        ;STORE VECTOR ADDRESS
3725  011250  012724  044444                MOV     #I.INTR,(R4)+   ;LOAD RK06 VECTOR ADDRESS
3726  011254  013714  001452                MOV     RKPRI,(R4)      ;  AND INTERRUPT PRIORITY
3727  011260  105737  001361          REST2: TSTB    FLAG
3728  011264  100022                        BPL     99$
3729  011266  122737  000013  000041        CMPB    #13.,@#41       ;LOAD FROM XXDP ?
3730  011274  001016                        BNE     99$             ;BRANCH IF NOT
3731  011276  104401  056224                TYPE    ,XXDPMG
3732  011302  000000                        HALT
3733  011304  122737  000013  000041        CMPB    #13.,@#41       ;DID THEY CHANGE THE PACK?
3734  011312  001007                        BNE     99$             ;CONTINUE IF THEY DID
3735  011314  112737  000000  056463        MOVB    #0,OPR001       ;LOAD DRIVE TO BE DROPPED
3736  011322  010446                        MOV     R4,-(SP)        ;SAVE R4
3737  011324  010546                        MOV     R5,-(SP)        ;SAVE R5
3738  011326  004737  013162                JSR     PC,DROP         ;GO DROP DRIVE 0
3739  011352                        99$:
```

```
 3740  011332  004737  043642                 JSR     PC,CLKINT       ;INITIALIZE CLOCK VECTOR ADDRESS
 3741  011336  104401  057772                 TYPE    ,SYS008         ;TYPE 'READY TO BEGIN TESTING'
 3742  011342  005227  177777                 INC     #-1             ;TYPE HELP MESSAGE ON FIRST PASS
 3743  011346  001002                         BNE     5$
 3744  011350  104401  066404                 TYPE    ,HELP
 3745  011354  004737  054002           5$:   JSR     PC,$TKINT       ;SET UP TTY INTERRUPT
 3746  011360  013737  001452  000036       MOV     RKPRI,TRAPVEC+2 ;ALLOW L AND P CLOCK INTERRUPTS
 3747  011366  005037  177776               CLR     PS              ;ALLOW RK06 INTERRUPTS
 3748
 3749  011372  122737  000002  001361       CMPB    #2,FLAG         ;SEE IF AUTO SIZE
 3750  011400  001401                         BEQ     1$              ;BR IF YES
 3751  011402  000406                         BR      MAIN            ;ELSE GOTO MAIN
 3752  011404  012746  011420           1$:   MOV     #MAIN,-(SP)     ;SETUP STACK TO RETRUN TO MAIN
 3753  011410  010446                         MOV     R4,-(SP)        ;SAVE R4
 3754  011412  010546                         MOV     R5,-(SP)        ;SAVE R5
 3755  011414  000137  012666                 JMP     WRTEPK+10               ;WRITE ALL DRIVES ON BUS & RET TO MAIN
 3756
```

```
                                        .SBTTL  MAIN IDLE LOOP

011420  022706  0C1100          MAIN:   CMP     #STACK,SP       ;CHECK IF STACK PROBLEM
011424  001401                          BEQ     1$              ;NO, CONTINUE
011426  000000                          HALT                    ; *** PROGRAM PROBLEM
011430  004737  052116          1$:     JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER
011434  004737  044270                  JSR     PC,W.WTCH       ;CALL WATCH DOG TIMER
011440  005737  001564                  TST     ERCONT          ;CHECK IF IN SPECIAL ERROR SEQUENCE
011444  100765                          BMI     MAIN            ;YES, DO NOT PRINT INTERVAL STATISTICS OR
                                                                ;  ADD DRIVES TO TESTING SEQUENCE
011446  032777  060000  167464          BIT     #SW14!SW13,@SWR ;CHECK IF LOOP ON OPERATION OR
                                                                ;  INHIBIT PRINT OUT
011454  001022                          BNE     11$             ;YES, DO NOT PRINT OUT INTERVAL STATISTICS
011456  105737  001633                  TSTB    STATIS          ;CHECK FOR INTERVAL STATISTICS
011462  001417                          BEQ     11$             ;NO, CONTINUE SCAN
011464  005004                          CLR     R4              ;START WITH DRIVE 0
011466  016405  001524          2$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
011472  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IS IN USE
011500  001402                          BEQ     3$              ;NO, LOOK AT NEXT PARAMETER BLOCK
011502  004737  013260                  JSR     PC,STAT         ;GET STATISTICS
011506  005724                  3$:     TST     (R4)+           ;INCREMENT DRIVE INDEX
011510  022704  000020                  CMP     #20,R4          ;CHECK IF ALL DRIVES SCANNED
011514  101364                          BHI     2$              ;NO, DO NEXT DRIVE
011516  105037  001633                  CLRB    STATIS          ;RESET FLAG

011522  004037  051644          11$:    JSR     R0,Q.POP        ;GET FIRST DRIVE FROM DRIVE
011526  001314                          AVAILQ                  ;  AVAILABLE QUEUE
011530  012605                          MOV     (SP)+,R5        ;STORE PARAMETER BLOCK ADDRESS
011532  001732                          BEQ     MAIN            ;IF QUEUE EMPTY WAIT
011534  032765  002000  000014          BIT     #DRPDRV,P.PRST(R5)  ;CHECK IF DROP DRIVE
011542  001403                          BEQ     12$             ;NO, GENERATE NEW COMMAND
011544  004737  013162                  JSR     PC,DROP         ;DROP DRIVE
011550  000723                          BR      MAIN            ;GET NEXT AVAILABLE DRIVE

011552  032777  040000  167360  12$:    BIT     #SW14,@SWR      ;CHECK IF LOOP ON OPERATION
011560  001002                          BNE     15$             ;YES, DO NOT CLEAR SEEK TO PREVIOUS
                                                                ;  CYLINDER FLAG
011562  105065  000211                  CLRB    P.SEEK(R5)      ;CLEAR SEEK TO PREVIOUS CYLINDER
011566  004737  021336          15$:    JSR     PC,GENERT       ;GENERATE NEW COMMAND
011572  105765  000211                  TSTB    P.SEEK(R5)      ;CHECK IF SEEK TO PREIOUS CYLINDER
011576  001310                          BNE     MAIN            ;YES, GET NEXT AVAILIABLE DRIVE
011600  013737  001452  177776          MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
011606  005737  001320                  TST     BWAITQ          ;CHECK IF BUFFER WAIT QUEUE EMPTY
011612  001013                          BNE     25$             ;NO, ENQUEUE PARAMETER BLOCK
011614  004037  022616                  JSR     R0,BUFFAL       ;ALLOCATE BUFFER
011620  011642                          25$                     ;NOT ENOUGH ROOM RETURN
011622  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
011626  004737  023234                  JSR     PC,LDDATA       ;LOAD BUFFER
011632  004037  051564                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON COMMAND
011636  001324                          CINITQ                  ;  INITIATION QUEUE
011640  000667                          BR      MAIN            ;GET NEXT AVAILABLE DRIVE

011642  004037  051564          25$:    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON BUFFER
011646  001320                          BWAITQ                  ;  WAIT QUEUE
011650  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
011654  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,JPRCMD
                                ;*      RETURN
                                ;*
                                ;*
                                ;*      COMMAND         ROUTINE
                                ;*      -------         -------
                                ;*
                                ;*      TN              TESTDV
                                ;*      PN              PARMDV
                                ;*      DN              DROPDV
                                ;*      SN              STATDV
                                ;*      WN              WRTEPK
                                ;*
                                ;;*****************************************************************

011656  010446                  OPRCMD: MOV     R4,-(SP)        ;STORE R4 ON STACK
011660  104401  056626                  TYPE    ,OPR007         ;TYPE "PLEASE ENTER COMMAND"
011664  104403                          RDLIN                   ;READ LINE FROM TTY
011666  012604                          MOV     (SP)+,R4        ;STORE ADDRESS OF INPUT STRING
011670  116437  000001  056463          MOVB    1(R4),OPR001    ;MOVE SECOND CHARACTER TO DRIVE
                                                                ;  REQUEST STORAGE
011676  122737  000101  056463          CMPB    #'A,OPR001      ;CHECK IF ALL DRIVES
011704  001421                          BEQ     4$              ;YES, PROCESS ALL DRIVES
011706  122737  000060  056463          CMPB    #'0,OPR001      ;CHECK IF LEGAL DRIVE NUMBER
011714  101004                          BHI     2$              ;NO, PRINT ERROR
011716  122737  000067  056463          CMPB    #'7,OPR001      ;CHECK IF 0-7
011724  103006                          BHIS    3$              ;YES, PROCESS DRIVE COMMAND
011726  104401  056446          2$:     TYPE    ,OPR000         ;TYPE "DRIVE NUMBER N ILLEGAL?"
011732  104401  056466                  TYPE    ,OPR002
011736  012604                          MOV     (SP)+,R4        ;RESTORE R4
011740  000207                          RTS     PC              ;RETURN

011742  142737  000370  056463  3$:     BICB    #370,OPR001     ;CLEAR UNIMPORTANT BITS
011750  010546                  4$:     MOV     R5,-(SP)        ;STORE R5 ON STACK
011752  122714  000124                  CMPB    #'T,(R4)        ;CHECK IF TEST DRIVE
011756  001427                          BEQ     TESTDV          ;YES, GO TO TEST DRIVE
011760  122714  000120                  CMPB    #'P,(R4)        ;CHECK IF TEST AND CHANGE PARAMETERS
011764  001512                          BEQ     PARMDV          ;YES, GO TEST AND CHANGE PARAMETERS
011766  122714  000104                  CMPB    #'D,(R4)        ;CHECK IF DROP DRIVE
011772  001553                          BEQ     DROPDV          ;YES, GO DROP DRIVE
011774  122714  000123                  CMPB    #'S,(R4)        ;CHECK IF DEMAND STATISTICS
012000  001002                          BNE     5$
012002  000137  012502                  JMP     STATDV          ;YES, GO TO DEMAND STATISTICS
```

```
 3869
 3870  012006  122714  000127          5$:     CMPB    #'W,(R4)        ;CHECK IF WRITE PACK AND TEST
 3871  012012  001002                          BNE     6$
 3872  012014  000137  012656                  JMP     WRTEPK          ;YES, GO WRITE PACK AND TEST
 3873
 3874  012020  111437  056440          6$:     MOVB    (R4),ILLCMD     ;STORE COMMAND FOR PRINT OUT
 3875  012024  104401  056407                  TYPE    ,ILLCOM         ;TYPE ERROR MESSAGE
 3876  012030  012605                          MOV     (SP)+,R5        ;RESTORE R5
 3877  012032  012604                          MOV     (SP)+,R4        ;RESTORE R4
 3878  012034  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  TEST DRIVE COMMAND
                                
012036  122737  000101  056463  TESTDV: CMPB    #'A,OPR001      ;CHECK IF ALL DRIVES ARE TO BE TESTED
012044  001027                          BNE     10$             ;NO, DO SPECIFIC DRIVE
012046  105037  001632                  CLRB    DRVCNT          ;CLEAR DRIVE COUNT
012052  005004                          CLR     R4              ;CLEAR DRIVE COUNT
012054  016405  001524          1$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
012060  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
012066  001004                          BNE     2$              ;YES, GO TO NEXT DRIVE
012070  004737  013536                  JSR     PC,TEST         ;GO TEST DRIVE
012074  105237  001632                  INCB    DRVCNT          ;INCREMENT DRIVE COUNT
012100  005724                  2$:     TST     (R4)+           ;ADDRESS NEXT DRIVE (ADD 2)
012102  022704  000020                  CMP     #20,R4          ;CHECK IF FINICHED
012106  001362                          BNE     1$              ;NO, SET UP NEXT DRIVE
012110  105737  001632                  TSTB    DRVCNT          ;CHECK IF ANY DRIVE IN ASSIGNMENT SEQUENCE
012114  001033                          BNE     15$             ;YES, RETURN
012116  104401  057005                  TYPE    ,OPR013         ;TYPE "ALL DRIVES CURRENTLY UNDER TEST?"
012122  000430                          BR      15$             ;RETURN
                                
012124  113704  056463          10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
012130  006304                          ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
012132  016405  001524                  MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
012136  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
012144  001003                          BNE     11$             ;YES, PRINT DRIVE ALREADY ASSIGNED
012146  004737  013536                  JSR     PC,TEST         ;GO TEST DRIVE
012152  000414                          BR      15$             ;RETURN
                                
012154  013737  001452  177776  11$:    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
012162  152737  000060  056463          BISB    #60,OPR001      ;MAKE NUMBER ASCIZ
012170  104401  056446                  TYPE    ,OPR000         ;TYPE 'DRIVE N ALREADY ASSIGNED?"
012174  104401  056514                  TYPE    ,OPR004
012200  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
012204  012605                  15$:    MOV     (SP)+,R5        ;RESTORE R5
012206  012604                          MOV     (SP)+,R4        ;RESTORE R4
012210  000207                          RTS     PC              ;RETURN
                                
                                        .SBTTL  CHANGE PARAMETERS COMMAND
                                
012212  122737  000101  056463  PARMDV: CMPB    #'A,OPR001      ;CHECK IF PARAMETERS FOR ALL DRIVES
012220  001020                          BNE     10$             ;NO, DO SPECIFIC DRIVE
012222  005004                          CLR     R4              ;CLEAR DRIVE COUNT
012224  016405  001524          1$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
012230  004737  016346                  JSR     PC,PARM         ;ASK FOR PARAMETERS
012234  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
012242  001002                          BNE     2$              ;YES, GO TO NEXT DRIVE
012244  004737  013536                  JSR     PC,TEST         ;GO TEST DRIVE
012250  005724                  2$:     TST     (R4)+           ;ADDRESS NEXT DRIVE (ADD 2)
012252  022704  000020                  CMP     #20,R4          ;CHECK IF FINISHED
012256  001362                          BNE     1$              ;NO, SET UP NEXT DRIVE
012260  000415                          BR      15$             ;RETURN
                                
012262  113704  056463          10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
012266  006304                          ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
012270  016405  001524                  MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
012274  004737  016346                  JSR     PC,PARM         ;GET NEW PARAMETERS
012300  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
```

```
                                                BNE     15$             ;YES, RETURN
                                                JSR     PC,TEST         ;GO TEST DRIVE
                                        15$:    MOV     (SP)+,R5        ;RESTORE R5
                                                MOV     (SP)+,R4        ;RESTORE R4
                                                RTS     PC              ;RETURN

                                        .SBTTL  DROP DRIVE COMMAND

                                        DROPDV: CMPB    #'A,OPR001      ;CHECK IF ALL DRIVES ARE TO BE DROPPED
                                                BNE     10$             ;NO, DO SPECIFIC DRIVE
                                                CLRB    DRVCNT          ;CLEAR DRIVE COUNT
                                                CLR     R4              ;CLEAR DRIVE COUNT
                                        1$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
                                                BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
                                                BEQ     2$              ;NO, GO TO NEXT DRIVE
                                                BIS     #DRPDRV,P.PRST(R5)  ;INDICATE DRIVE IS TO BE DROPPED
                                                INCB    DRVCNT          ;INCREMENT DRIVE COUNT
                                        2$:     TST     (R4)+           ;ADDRESS NEXT DRIVE (ADD 2)
                                                CMP     #20,R4          ;CHECK IF FINISHED
                                                BNE     1$              ;NO, SET UP NEXT DRIVE
                                                TSTB    DRVCNT          ;CHECK IF ANY DRIVES AVAILIABLE
                                                BNE     15$             ;YES, RETURN
                                                TYPE    ,OPR014         ;TYPE 'NO DRIVES IN USE?'
                                                BR      15$             ;RETURN

                                        10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
                                                ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
                                                MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
                                                BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
                                                BEQ     11$             ;NO, PRINT DRIVE N NOT ASSIGNED
                                                BIS     #DRPDRV,P.PRST(R5)  ;INDICATE DRIVE IS TO BE DROPPED
                                                BR      15$             ;RETURN

                                        11$:    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
                                                BISB    #60,OPR001      ;MAKE NUMBER ASCII
                                                TYPE    ,OPR000         ;TYPE 'DRIVE N NOT ASSIGNED?'
                                                TYPE    ,OPR005
                                                CLR     PS              ;ALLOW RK06 INTERRUPTS
                                        15$:    MOV     (SP)+,R5        ;RESTORE R5
                                                MOV     (SP)+,R4        ;RESTORE R4
                                                RTS     PC              ;RETURN

                                        .SBTTL  GET DRIVE STATISTICS COMMAND

                                        STATDV: CMPB    #'A,OPR001      ;CHECK IF STATISTICS FOR ALL DRIVES
                                                BNE     10$             ;NO, DP SPECIFIC DRIVE
                                                CLRB    DRVCNT          ;CLEAR DRIVE COUNT
                                                CLR     R4              ;CLEAR DRIVE COUNT
                                        1$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
                                                BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
                                                BEQ     2$              ;NO, GO TO NEXT DRIVE
                                                JSR     PC,STAT         ;GATHER STATISTICS
                                                INCB    DRVCNT          ;INCREMENT DRIVE COUNT
                                        2$:     TST     (R4)+           ;ADDRESS NEXT DRIVE (ADD 2)
                                                CMP     #20,R4          ;CHECK IF FINISHED
                                                BNE     1$              ;NO, SET UP NEXT DRIVE
```

```
                                                TSTB    DRVCNT          ;CHECK IF ANY DRIVES AVAILIABLE
                                                BNE     15$             ;YES, RETURN
                                                TYPE    ,OPR014         ;TYPE 'NO DRIVES IN USE?'
                                                BR      15$             ;RETURN

                                        10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
                                                ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
                                                MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
                                                BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
                                                BEQ     11$             ;PRINT DRIVE N NOT ASSIGNED
                                                JSR     PC,STAT         ;GO GATHER DRIVE STATISTICS
                                                BR      15$             ;RETURN

                                        11$:    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
                                                BISB    #60,OPR001      ;MAKE NUMBER ASCII
                                                TYPE    ,OPR000         ;TYPE 'DRIVE N NOT ASSIGNED?'
                                                TYPE    ,OPR005
                                                CLR     PS              ;ALLOW RK06 INTERRUPTS
                                        15$:    MOV     (SP)+,R5        ;RESTORE R5
                                                MOV     (SP)+,R4        ;RESTORE R4
                                                RTS     PC              ;RETURN

                                        .SBTTL  WRITE PACK AND TEST COMMAND

                                        WRTEPK: CMPB    #'A,OPR001      ;CHECK IF ALL DRIVES ARE TO BE TESTED
                                                BNE     10$             ;NO, DO SPECIFIC DRIVE
                                                CLRB    DRVCNT          ;CLEAR DRIVE COUNT
                                                CLR     R4              ;CLEAR DRIVE COUNT
                                        1$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
                                                BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
                                                BNE     2$              ;YES, GO TO NEXT DRIVE
                                                TSTB    P.DRVN(R5)
                                                BNE     97$
                                                CMPB    #13,@#41        ;LOAD FROM XXDP ?
                                                BNE     97$             ;BRANCH IF NOT
                                                TYPE    ,XXDPMG
                                                HALT
                                                CMPB    #13,@#41        ;DID THEY CHANGE THE PACK?
                                                BNE     97$             ;CONTINUE IF THEY DID
                                                BISB    #BIT0,P.ASSN(R5);SET UP FOR NO STATS
                                                JSR     PC,DROP         ;GO DROP DRIVE 0
                                        97$:    MOVB    #BIT4,P.ASSN(R5) ;SET FLAG FOR WRITE PACK
                                                JSR     PC,TEST         ;GO ASSIGN DRIVE
                                                INCB    DRVCNT          ;INCREMENT DRIVE COUNT
                                        2$:     TST     (R4)+           ;ADDRESS NEXT DRIVE (ADD 2)
                                                CMP     #20,R4          ;CHECK IF FINISHED
                                                BNE     1$              ;NO, SET UP NEXT COMMAND
                                                TSTB    DRVCNT          ;CHECK IF ANY DRIVES NOT IN USE
                                                BEQ     15$             ;NO, RETURN
                                                TYPE    ,OPR013         ;TYPE "ALL DRIVES CURRENTLY UNDER TEST"
                                                BR      15$             ;RETURN

                                        10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
                                                ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
                                                MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
                                                BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
```

```
 4047  013040  001031                        BNE     11$             ;YES, PRINT DRIVE ALREADY ASSIGNED
 4048  013042  105765  000000                TSTB    P.DRVN(R5)
 4049  013046  001020                        BNE     99$
 4050  013050  122737  000013  000041        CMPB    #13,@#41        ;LOAD FROM XXDP ?
 4051  013056  001014                        BNE     99$             ;BRANCH IF NOT
 4052  013060  104401  056224                TYPE    ,XXDPMG
 4053  013064  000000                        HALT
 4054  013066  122737  000013  000041        CMPB    #13,@#41        ;DID THEY CHANGE THE PACK?
 4055  013074  001005                        BNE     99$             ;CONTINUE IF THEY DID
 4056  013076  152765  000001  000210        BISB    #BIT0,P.ASSN(R5);SET UP FOR NO STAT
 4057  013104  004737  013162                JSR     PC,DROP         ;GO DROP DRIVE 0
 4058  013110  112765  000020  000210  99$:   MOVB    #BIT4,P.ASSN(R5) ;SET FLAG FOR WRITE PACK
 4059  013116  004737  013536                JSR     PC,TEST         ;GO ASSIGN DRIVE
 4060  013122  000414                        BR      15$             ;RETURN
 4061
 4062  013124  013737  001452  177776  11$:   MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 4063  013132  152737  000060  056463        BISB    #60,OPR001      ;MAKE NUMBER ASCII
 4064  013140  104401  056446                TYPE    ,OPR000         ;TYPE 'DRIVE N ALREADY ASSIGNED?"
 4065  013144  104401  056514                TYPE    ,OPR004
 4066  013150  005037  177776                CLR     PS              ;ALLOW RK06 INTERRUPTS
 4067  013154  012605                15$:    MOV     (SP)+,R5        ;RESTORE R5
 4068  013156  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4069  013160  000207                        RTS     PC              ;RETURN
 4070
```

```
                                        .SBTTL  DROP DRIVE AND GATHER STATISTICS ROUTINES

013162  013746  177776          DROP:   MOV     PS,-(SP)        ;STORE PSW
013166  013737  001452  177776          MOV     RKPRI,PS        ;LOCK OUT TTY AND RK06 INTERRUPTS
013174  005065  000014                  CLR     P.PRST(R5)      ;CLEAR PROGRAM STAUS REGISTER
013200  005065  000212                  CLR     P.ERR(R5)       ;CLEAR ERROR FLAGS
013204  116537  000000  056755          MOVB    P.DRVN(R5),OPRC11  ;LOAD DRIVE NUMBER
013212  152737  000060  056755          BISB    #60,OPR011      ;MAKE IT ASCII
013220  104401  056736                  TYPE    ,OPR010         ;TYPE 'DRIVE N HAS BEEN DROPPED
013224  104401  056560                  TYPE    ,OPR006         ;  FROM TEST SEQUENCE''
013230  132765  000001  000210          BITB    #BIT0,P.ASSN(R5) ;SEE IF SIZING FOR DRIVE
013236  001404                          BEQ     2$              ;BR IF NOT
013240  105065  000210                  CLRB    P.ASSN(R5)      ;CLEAR ASSINGMENT CODE BITS
013244  000137  013530                  JMP     STAT01
013250  105065  000210          2$:     CLRB    P.ASSN(R5)      ;CLEAR ASSINGMENT CODE BITS
013254  000137  013272                  JMP     STAT00

013260  013746  177776          STAT:   MOV     PS,-(SP)        ;STORE PSW
013264  013737  001452  177776          MOV     RKPRI,PS        ;LOCK OUT TTY AND RK06 INTERRUPTS
013272  104401  057124          STAT00: TYPE    ,DRVSTT         ;**DRIVE STATISTICS
013276  116537  000000  056755          MOVB    P.DRVN(R5),OPR011  ;LOAD DRIVE NUMBER FOR PRINT OUT
013304  152737  000060  056755          BISB    #60,OPR011      ;MAKE IT ASCII
013312  104401  056736                  TYPE    ,OPR010         ;TYPE 'DRIVE N''
013316  004737  027774                  JSR     PC,PRISER       ;PRINT DRIVE AND PACK SERIAL NUMBERS
013322  004737  044046                  JSR     PC,PRITIM       ;PRINT TIME IF CLOCK AVAILIABLE
013326  104401  057202                  TYPE    ,STT001         ;TYPE 'ORDERS PERFORMED''
013332  010546                          MOV     R5,-(SP)        ;GET PARAMETER BLOCK ADDRESS
013334  062716  000152                  ADD     #P.NODR,(SP)    ;CALCULATE ADDRESS FOR PRINT OUT
013340  004737  055470                  JSR     PC,$DB2D        ;CONVERT TO ASCII
013344  104401  057224                  TYPE    ,STT002         ;TYPE 'WORDS WRITTEN*65K''
013350  010546                          MOV     R5,-(SP)        ;GET PARAMETER BLOCK ADDRESS
013352  062716  000160                  ADD     #P.NWRT+2,(SP)  ;CALCULATE ADDRESS FOR PRINT OUT
013356  004737  055470                  JSR     PC,$DB2D        ;CONVERT TO ASCII
013362  104401  057251                  TYPE    ,STT003         ;TYPE 'WORDS READ*65K''
013366  010546                          MOV     R5,-(SP)        ;GET PARAMETER BLOCK ADDRESS
013370  062716  000166                  ADD     #P.NRD+2,(SP)   ;CALCULATE ADDRESS FOR PRINT OUT
013374  004737  055470                  JSR     PC,$DB2D        ;CONVERT TO ASCII
013400  104401  057273                  TYPE    ,STT004         ;TYPE ''SOFT DATA ERRORS''
                                        ;  'ECC''
013404  016546  000176                  MOV     P.SECC(R5),-(SP) ;STORE SOFT ECC ERROR COUNT
013410  004737  055426                  JSR     PC,$SB2D        ;CONVERT TO DECIMAL
013414  104401  057326                  TYPE    ,STT005         ;TYPE 'REREAD''
013420  016546  000172                  MOV     P.SRRD(R5),-(SP) ;SAVE SOFT REREAD CORRECTABLE
013424  004737  055426                  JSR     PC,$SB2D        ;CONVERT TO DECIMAL
013430  104401  057342                  TYPE    ,STT006         ;TYPE 'OFFSET''
013434  016546  000174                  MOV     P.SOFF(R5),-(SP) ;SAVE SOFT OFFSET CORRECTABLE
013440  004737  055426                  JSR     PC,$SB2D        ;CONVERT TO DECIMAL
013444  104401  057356                  TYPE    ,STT007         ;TYPE ''HARD DATA ERRORS''
013450  016546  000200                  MOV     P.HARD(R5),-(SP) ;SAVE HARD DATA ERRORS
013454  004737  055426                  JSR     PC,$SB2D        ;CONVERT TO DECIMAL
013460  104401  057402                  TYPE    ,STT008         ;TYPE ''SEEK INCOMPLETES
013464  016546  000202                  MOV     P.NSKI(R5),-(SP) ;TYPE NUMBER OF SEEK INCOMPLETES
013470  004737  055426                  JSR     PC,$SB2D        ;CONVERT TO DECIMAL
013474  104401  057426                  TYPE    ,STT009         ;TYPE ''OPERATION INCOMPLETES''
013500  016546  000204                  MOV     P.NOPI(R5),-(SP) ;SAVE NUMBER OF OPERATION INCOMPLETES
013504  004737  055426                  JSR     PC,$SB2D        ;CONVERT TO DECIMAL
```

```
 4127  013510  104401  057474                  TYPE    ,STT010           ;TYPE 'OTHER ERRORS''
 4128  013514  016546  000206                  MOV     P.NER(R5),-(SP)   ;SAVE NUMBER OF OTHER ERRORS
 4129  013520  004737  055426                  JSR     PC,$SB2D          ;CONVERT TO DECIMAL
 4130  013524  104401  001165                  TYPE    ,$CRLF            ;TYPE <CR><LF>
 4131  013530  012637  177776          STAT01: MOV     (SP)+,PS          ;RESTORE PSW
 4132  013534  000207                          RTS     PC                ;RETURN
 4133
```

```
                                        .SBTTL  INITIATE DRIVE ASSIGNMENT SEQUENCE
                                
013536  116546  000210        TEST:   MOVB    P.ASSN(R5),-(SP) ;SAVE ASSIGNMENT CODE
013542  010446                        MOV     R4,-(SP)         ;STORE R4 ON STACK
013544  010346                        MOV     R3,-(SP)         ;STORE R3 ON STACK
013546  010503                        MOV     R5,R3            ;LOAD PARAMETER BLOCK ADDRESS INTO R3
013550  062703  000122                ADD     #P.RDPT,R3       ;CALCULATE BEGINNING OF STATISTICS AND
                                                                ;  ERROR AREA
013554  010504                        MOV     R5,R4            ;LOAD PARAMETER BLOCK ADDRESS INTO R4
013556  062704  000232                ADD     #P.EXAR,R4       ;CALCULATE END OF STATISTICS AND
                                                                ;  ERROR AREA
013562  005023                1$:     CLR     (R3)+            ;CLEAR STATISTIC AND ERROR AREA
013564  020304                        CMP     R3,R4            ;CHECK IF FINISHED
013566  001375                        BNE     1$               ;NO, CONTINUE
013570  116504  000000                MOVB    P.DRVN(R5),R4    ;STORE DRIVE NUMBER
013574  156437  001514  001513        BISB    INTMSK(R4),O.OVER ;SET UP FOR OVERLAPPING SEEKS
013602  012603                        MOV     (SP)+,R3         ;RESTORE R3
013604  012604                        MOV     (SP)+,R4         ;RESTORE R4
013606  112665  000210                MOVB    (SP)+,P.ASSN(R5) ;RESTORE ASSIGNMENT CODE
                                
013612  105065  000117        TEST1:  CLRB    P.ECMP(R5)       ;CLEAR ECC COMPARE FLAG
013616  005037  001500                CLR     OPTFLG           ;CLEAR CMND OPTIMIZER FLAG
013622  152765  000001  000210        BISB    #BIT0,P.ASSN(R5) ;SET READ STATUS COMMAND FLAG
013630  052765  000001  000014        BIS     #DRVUSE,P.PRST(R5) ;SET DRIVE IN USE
013636  112765  000141  000001        MOVB    #RDSTAT,P.CMND(R5) ;READ ALL DRIVE STATUS
013644  112765  000141  000123        MOVB    #RDSTAT,P.RCMD(R5)
013652  004037  051564                JSR     R0,Q.PUSH        ;ENQUEUE PARAMETER BLOCK IN
013656  001324                        CINITQ                   ;  COMMAND INITIATION QUEUE
013660  000207                        RTS     PC               ;RETURN
```

```
                                        .SBTTL  DRIVE ASSIGNMENT SEQUENCE
                                
013662  132765  000001  000210  ASNORM: BITB    #BIT0,P.ASSN(R5) ;CHECK IF READ STATUS
013670  001467                          BEQ     ASN4$            ;NO, CHECK IF START SPINDLE
013672  005737  027334                  TST     NEWFLG           ;SEE IF DTYE ERROR ON ASSIGNING
013676  001404                          BEQ     1$               ;BR IF NO
013700  005037  027334                  CLR     NEWFLG           ;ELSE CLEAR DTYE ON ASSIGN FLAG
013704  000137  013612                  JMP     TEST1            ;& REPEAT SELECT FOR DRIVE THAT HAD DTYE
013710  142765  000001  000210  1$:     BICB    #BIT0,P.ASSN(R5) ;CLEAR READ STATUS COMMAND ISSUED
013716  016565  000054  000224          MOV     P.A11(R5),P.SERL(R5) ;STORE SERIAL NUMBER
013724  042765  100007  000224          BIC     #^C<M.SER>,P.SERL(R5) ;KEEP SERIAL NUMBER
013732  006265  000224                  ASR     P.SERL(R5)       ;SHIFT 3 BITS RIGHT
013736  006265  000224                  ASR     P.SERL(R5)
013742  006265  000224                  ASR     P.SERL(R5)
013746  032765  000200  000036          BIT     #DRDY,P.DS(R5)   ;CHECK IF DRIVE READY
013754  001415                          BEQ     ASN2$            ;NO, ISSUE START SPINDLE
013756  116565  000123  000135  ASN1$:  MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
013764  112765  000103  000001          MOVB    #PACK,P.CMND(R5) ;ISSUE PACK ACKNOWLEDGE
013772  112765  000103  000123          MOVB    #PACK,P.RCMD(R5)
014000  152765  000004  000210          BISB    #BIT2,P.ASSN(R5) ;SET PACK ACKNOWLEDGE FLAG
014006  000414                          BR      ASN3$            ;ISSUE PACK ACKNOWLEDGE AND WAIT
                                
014010  116565  000123  000135  ASN2$:  MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
014016  112765  000111  000001          MOVB    #SRTSPL,P.CMND(R5) ;ISSUE START SPINDLE
014024  112765  000111  000123          MOVB    #SRTSPL,P.RCMD(R5)
014032  152765  000002  000210          BISB    #BIT1,P.ASSN(R5) ;SET START SPINDLE FLAG
014040  004037  051564          ASN3$:  JSR     R0,Q.PUSH        ;ENQUEUE PARAMETER BLOCK IN
014044  001324                          CINITQ                   ;  COMMAND INITIATION QUEUE
014046  000207                          RTS     PC               ;RETURN
                                
014050  132765  000002  000210  ASN4$:  BITB    #BIT1,P.ASSN(R5) ;CHECK IF START SPINDLE
014056  001414                          BEQ     15$              ;NO, CHECK IF PACK ACKNOWLEDGE
014060  142765  000002  000210          BICB    #BIT1,P.ASSN(R5) ;CLEAR START SPINDLE FLAG
014066  032765  000200  000036          BIT     #DRDY,P.DS(R5)   ;CHECK IF DRIVE READY
014074  001330                          BNE     ASN1$            ;YES, ISSUE PACK ACKNOWLEDGE
014076  052765  002000  000212          BIS     #BIT10,P.ERR(R5) ;SET DRIVE NOT READY AFTER START SPINDLE
014104  000137  025470                  JMP     ABNORM           ;REPORT ERROR
                                
014110  132765  000004  000210  15$:    BITB    #BIT2,P.ASSN(R5) ;CHECK IF PACK ACKNOWLEDGE
014116  001431                          BEQ     18$              ;NO, CHECK IF RECAL
014120  142765  000004  000210          BICB    #BIT2,P.ASSN(R5) ;CLEAR PACK ACKNOWLEDGE FLAG
014126  032765  000100  000036          BIT     #VV,P.DS(R5)     ;CHECK IF VOLUME VALID IS SET
014134  001005                          BNE     16$              ;YES, CHECK IF PACK IS TO BE WRITTEN
014136  052765  004000  000212          BIS     #BIT11,P.ERR(R5) ;SET VOLUME VALID NOT SET AFTER PACK
                                                                 ;  ACKNOWLEDGE
014144  000137  025470                  JMP     ABNORM           ;REPORT ERROR
                                
014150  116565  000123  000135  16$:    MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
014156  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;ISSUE RECAL
014164  112765  000113  000123          MOVB    #RECAL,P.RCMD(R5)
014172  152765  000100  000210          BISB    #BIT6,P.ASSN(R5) ;SET RECAL ISSUED
014200  000717                          BR      ASN3$
                                
014202  132765  000100  000210  18$:    BITB    #BIT6,P.ASSN(R5) ;CHECK IF RECAL
014210  001457                          BEQ     23$              ;NO, CHECK IF READ PACK SERIAL NUMBER
014212  142765  000100  000210          BICB    #BIT6,P.ASSN(R5) ;RESET RECAL ISSUED
```

```
4219  014220  152765  000010  000210          BISB    #BIT3,P.ASSN(R5) ;SET ASSIGNMENT CODE FOR PACK
4220                                                    ;  SERIAL NUMBER READ
4221  014226  012765  001000  000004          MOV     #1000,P.SECT(R5) ;  TRACK 2 SECTOR 0
4222  014234  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
4223  014242  112765  000121  000001  20$:    MOVB    #RDDATA,P.CMND(R5) ;LOAD READ COMMAND FOR PACK
4224  014250  112765  000121  000123          MOVB    #RDDATA,P.RCMD(R5) ;  SERIAL NUMBER
4225  014256  012765  177774  000012          MOV     #-4,P.WC(R5)    ;LOAD WORD COUNT FOR THE FIRST FOUR WORDS
4226  014264  004737  027252                  JSR     PC,GETNUM       ;GET CORRECT END CYL NUM
4227  014270  013765  027332  000002          MOV     LSTCYL,P.CYLN(R5) ;SET CYLINDER TO 632/1456
4228  014276  013765  027332  000124          MOV     LSTCYL,P.RCYL(R5)
4229  014304  016565  000004  000126          MOV     P.SECT(R5),P.RSEC(R5) ;STORE TRACK AND SECTOR
4230  014312  005737  001320                  TST     BWAITQ          ;CHECK IF BUFFER WAIT QUEUE IS EMPTY
4231  014316  001010                          BNE     22$             ;NO, ENQUEUE PARAMETER BLOCK IN BUFFER
4232                                                                  ;  WAIT QUEUE
4233  014320  004037  022616                  JSR     R0,BUFFAL       ;ALLOCATE BUFFER
4234  014324  014340                          22$                     ;NOT ENOUGH ROOM RETURN
4235  014326  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4236  014332  004737  023234                  JSR     PC,LDDATA       ;CLEAR BUFFER
4237  014336  000640                          BR      ASN3$           ;ISSUE COMMAND
4238
4239  014340  004037  051564          22$:    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON BUFFER
4240  014344  001320                          BWAITQ                  ;  WAIT QUEUE
4241  014346  000207                          RTS     PC              ;RETURN
4242
4243  014350  032777  040000  164562  23$:    BIT     #SW14,@SWR      ;CHECK IS CYCLE ON OPERATION
4244  014356  001542                          BEQ     43$             ;NO, CONTINUE WITH ASSIGNMENT PROCESS
4245  014360  105765  000211                  TSTB    P.SEEK(R5)      ;CHECK IF SEEK TO PREVIOUS POSITION
4246  014364  001444                          BEQ     28$             ;NO, SERVICE INTERRUPT
4247  014366  105065  000211                  CLRB    P.SEEK(R5)      ;CLEAR FLAG
4248  014372  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE LAST COMMAND
4249  014400  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5) ;LOAD CYLINDER ADDRESS
4250  014406  016565  000126  000004          MOV     P.RSEC(R5),P.SECT(R5) ;LOAD PREVIOUS TRACK AND SECTOR
4251  014414  016565  000130  000012          MOV     P.RWC(R5),P.WC(R5) ;LOAD PREVIOUS WORD COUNT
4252  014422  116565  000122  000121          MOVB    P.RDPT(R5),P.DPAT(R5) ;LOAD PREVIOUS PATTERN
4253  014430  132765  000010  000210          BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
4254  014436  001003                          BNE     25$             ;YES, DO NOT SET WRITE BEFORE WRITE CHECK
4255  014440  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;SET WRITE BEFORE WRITE CHECK
4256  014446  005737  001320          25$:    TST     BWAITQ          ;CHECK IF ANY DRIVE WAITING FOR BUFFER
4257  014452  001332                          BNE     22$             ;YES, ENQUEUE PARAMETER BLOCK
4258  014454  004037  022616                  JSR     R0,BUFFAL       ;GO ALLOCATE BUFFER
4259  014460  014340                          22$                     ;NOT ENOUGH ROOM RETURN
4260  014462  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4261  014466  004737  023234                  JSR     PC,LDDATA       ;GO LOAD DATA
4262  014472  000137  014040                  JMP     ASN3$           ;GO ISSUE COMMAND
4263
4264  014476  004037  025210          28$:    JSR     R0,CHKADD       ;CHECK SECTOR, TRACK, CYLINDER,
4265                                                                  ;  WORD COUNT, AND BUS ADDRESS
4266  014502  015410                          60$                     ;ERROR RETURN
4267  014504  132765  000010  000210          BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
4268  014512  001424                          BEQ     35$             ;NO, RELEASE BUFFER
4269  014514  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
4270  014516  016504  000010                  MOV     P.BALO(R5),R4   ;LOAD R4 TO GET PACK SERIAL NUMBER
4271  014522  012465  000226                  MOV     (R4)+,P.PKSR(R5) ;GET PACK SERIAL NUMBER
4272  014526  012465  000230                  MOV     (R4)+,P.PKSR+2(R5)
4273  014532  005724                          TST     (R4)+           ;ADJUST R4 TO ADDRESS ALLIGNMENT INDICATION
4274  014534  005714                          TST     (R4)            ;CHECK IF DATA PACK
```

```
 4275  014536  001423                        BEQ     37$             ;YES, CONTINUE
 4276  014540  012604                30$:    MOV     (SP)+,R4        ;RESTORE R4
 4277  014542  004737  023020                JSR     PC,BUFREL       ;RELEASE BUFFER
 4278  014546  104401  057071                TYPE    ,OPR015         ;TYPE "ALLIGNMENT PACT IN DRIVE"
 4279  014552  004737  013162                JSR     PC,DROP         ;GO DROP DRIVE
 4280  014556  004737  022520                JSR     PC,GETBUF       ;GO GET BUFFER FOR NEXT COMMAND
 4281  014562  000207                        RTS     PC              ;RETURN
 4282
 4283  014564  032765  000200  000014  35$:  BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK HAS BEEN ISSUED
 4284  014572  001406                        BEQ     38$             ;YES, GO RELEASE BUFFER
 4285  014574  042765  000200  000014        BIC     #W.WCK,P.PRST(R5) ;CLEAR WRITE BEFORE WRITE CHECK
 4286  014602  000137  014040                JMP     ASN3$           ;GO ISSUE COMMAND
 4287
 4288  014606  012604                37$:    MOV     (SP)+,R4        ;RESTORE R4
 4289  014610  004737  023020          38$:  JSR     PC,BUFREL       ;RELEASE BUFFER
 4290  014614  032765  002000  000014        BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
 4291  014622  001402                        BEQ     40$             ;NO, SEEK TO PREVIOUS POSITION
 4292  014624  000137  016216                JMP     WV1$            ;GO DROP DRIVE
 4293
 4294  014630  112765  177777  000211  40$:  MOVB    #-1,P.SEEK(R5)  ;SET FLAG
 4295  014636  016565  000136  000002        MOV     P.LCYL(R5),P.CYLN(R5) ;LOAD LAST CYLINDER
 4296  014644  016565  000140  000004        MOV     P.LSEC(R5),P.SECT(R5) ;LOAD LAST SECTOR AND TRACK
 4297  014652  112765  000117  000001        MOVB    #SEEK,P.CMND(R5) ;ISSUE SEEK
 4298  014660  000137  014040                JMP     ASN3$           ;GO SEEK TO PREVIOUS POSITION
 4299
 4300  014664  132765  000010  000210  43$:  BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
 4301  014672  001002                        BNE     45$             ;YES, SET UP FOR READ
 4302  014674  000137  015412                JMP     WRTVER          ;CHECK IF WRITE PACK
 4303
 4304  014700  132765  000200  000210  45$:  BITB    #BIT7,P.ASSN(R5) ;CHECK IF RECAL AND RETRY
 4305                                                                ;  READ OF PACK SERIAL NUMBER
 4306  014706  001427                        BEQ     48$             ;NO, SERVICE READ OF PACK SERIAL NUMBER
 4307  014710  142765  000200  000210        BICB    #BIT7,P.ASSN(R5) ;RESET RECAL FOR REREAD OF PACK SERIAL NO.
 4308  014716  116565  000123  000135        MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE LAST COMMAND
 4309  014724  016565  000124  000136        MOV     P.RCYL(R5),P.LCYL(R5) ;STORE LAST CYLINDER
 4310  014732  016565  000126  000140        MOV     P.RSEC(R5),P.LSEC(R5) ;STORE LAST SECTOR AND TRACK
 4311  014740  016565  000130  000142        MOV     P.RWC(R5),P.LWC(R5) ;STORE LAST WORD COUNT
 4312  014746  062765  000002  000126        ADD     #2,P.RSEC(R5)   ;TRY NEXT 16-BIT FORMAT SECTOR
 4313  014754  016565  000126  000004        MOV     P.RSEC(R5),P.SECT(R5) ;STORE SECOTOR TO BE READ
 4314  014762  000137  014242                JMP     20$             ;GO ISSUE READ OF PACK SERIAL NUMBER
 4315
 4316  014766  142765  000010  000210  48$:  BICB    #BIT3,P.ASSN(R5) ;CLEAR FLAG
 4317  014774  004037  025210                JSR     R0,CHKADD       ;CHECK IF DATA TRANSFER CORRECT
 4318  015000  015410                        60$                     ;ERROR RETURN
 4319  015002  010446                        MOV     R4,-(SP)        ;STORE R4
 4320  015004  016504  000132                MOV     P.RBAL(R5),R4   ;LOAD R4 TO ADDRESS SERIAL NUMBER
 4321  015010  012465  000226                MOV     (R4)+,P.PKSR(R5) ;LOAD LEAST SIGNIFICANT BITS
 4322                                                                ;  OF SERIAL NUMBER
 4323  015014  012465  000230                MOV     (R4)+,P.PKSR+2(R5) ;LOAD MOST SIGNIFICANT BITS
 4324                                                                ;  OF SERIAL NUMBER
 4325  015020  005724                        TST     (R4)+           ;ADJUST R4 TO ADDRESS ALLIGNMENT INDICATION
 4326  015022  005714                        TST     (R4)            ;CHECK IF ALLIGNMENT PACK
 4327  015024  001245                        BNE     30$             ;YES, DROP DRIVE
 4328  015026  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4329  015030  004737  023020                JSR     PC,BUFREL       ;RELEASE BUFFER
 4330  015034  132765  000020  000210        BITB    #BIT4,P.ASSN(R5) ;CHECK IF WRITE PACK
```

```
4331  015042  001042                        BNE     55$             ;YES, SET UP PATTERN AND ALLOCATE BUFFER
4332  015044  105765  000076                TSTB    P.RATE(R5)      ;CHECK IF READ ONLY MODE
4333  015050  001413                        BEQ     52$             ;YES, START TESTING
4334  015052  032765  004000  000036        BIT     #WRL,P.DS(R5)   ;CHECK IF WRITE LOCKED
4335  015060  001407                        BEQ     52$             ;NO, START TESTING
4336
4337  015062  104401  056760          50$:  TYPE    ,OPR012         ;TYPE 'DRIVE WRITE LOCKED'
4338  015066  004737  013162                JSR     PC,DROP         ;DROP DRIVE FROM TEST SEQUENCE
4339  015072  004737  022520                JSR     PC,GETBUF       ;GET BUFFER FOR NEXT COMMAND
4340  015076  000207                        RTS     PC              ;RETURN
4341
4342  015100  116537  000000  056755  52$:  MOVB    P.DRVN(R5),OPR011 ;LOAD DRIVE NUMBER
4343  015106  152737  000060  056755        BISB    #60,OPR011      ;MAKE IT ASCII
4344  015114  104401  056736                TYPE    ,OPR010         ;TYPE 'DRIVE NUMBER N UNDER TEST'
4345  015120  104401  056501                TYPE    ,OPR003
4346  015124  004737  027774                JSR     PC,PRISER       ;PRINT DRIVE AND PACK SERIAL NUMBERS
4347  015130  004737  044046                JSR     PC,PRITIM       ;PRINT TIME
4348  015134  104401  001165                TYPE    ,$CRLF
4349  015140  004037  051564                JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN DRIVE
4350  015144  001314                        AVAILQ                  ;  AVAILIABLE QUEUE
4351  015146  000207                        RTS     PC              ;RETURN
4352
4353                                  .SBTTL  WRITE WHOLE PACK WITH RANDOM DATA
4354
4355  015150  032765  004000  000036  55$:  BIT     #WRL,P.DS(R5)   ;CHECK IF WRITE LOCKED
4356  015156  001341                        BNE     50$             ;YES, DROP DRIVE
4357  015160  005037  177776                CLR     PS              ;ALLOW RK06 INTERRUPTS
4358  015164  116565  000123  000135        MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE LAST COMMAND
4359  015172  016565  000124  000136        MOV     P.RCYL(R5),P.LCYL(R5) ;STORE LAST CYLINDER
4360  015200  016565  000126  000140        MOV     P.RSEC(R5),P.LSEC(R5) ;STORE LAST TRACK AND SECTOR
4361  015206  016565  000130  000142        MOV     P.RWC(R5),P.LWC(R5) ;STORE LAST WORD COUNT
4362  015214  112765  000131  000001        MOVB    #WRTCHK,P.CMND(R5) ;LOAD WRITE COMMAND
4363  015222  112765  000131  000123        MOVB    #WRTCHK,P.RCMD(R5)
4364  015230  052765  000200  000014        BIS     #W.WCK,P.PRST(R5) ;SET WRITE BEFORE WRITE CHECK
4365  015236  005065  000002                CLR     P.CYLN(R5)      ;CLEAR CYLINDER ADDRESS
4366  015242  005065  000124                CLR     P.RCYL(R5)
4367  015246  005065  000004                CLR     P.SECT(R5)      ;CLEAR TRACK AND SECTOR
4368  015252  005065  000126                CLR     P.RSEC(R5)
4369  015256  105065  000121                CLRB    P.DPAT(R5)      ;LOAD RANDOM PATTERN
4370  015262  013765  001354  000012        MOV     MAXBUF,P.WC(R5) ;LOAD WORD COUNT
4371  015270  042765  000377  000012        BIC     #377,P.WC(R5)   ;MAKE IT AN EVEN NUMBER OF SECTORS
4372  015276  001003                        BNE     57$             ;IF NOT ZERO USE THIS AS WORD COUNT
4373  015300  012765  000400  000012        MOV     #256.,P.WC(R5)  ;TRANSFER AT LEAST 256 WORDS
4374  015306  062765  000002  000152  57$:  ADD     #2,P.NODR(R5)   ;INCREMENT NUMBER OF ORDERS
4375  015314  066565  000012  000156        ADD     P.WC(R5),P.NWRT(R5) ;ADD NUMBER OF WORDS WRITTEN
4376  015322  005565  000160                ADC     P.NWRT+2(R5)
4377  015326  066565  000012  000164        ADD     P.WC(R5),P.NRD(R5) ;ADD NUMBER OF WORDS READ
4378  015334  005565  000166                ADC     P.NRD+2(R5)
4379  015340  005465  000012                NEG     P.WC(R5)        ;MAKE THE WORD COUNT NEGITIVE
4380  015344  013737  001452  177776        MOV     RKPRI,PS        ;LOCK RK06 INTERRUPTS
4381  015352  005737  001320                TST     BWAITQ          ;CHECK IF COMMANDS WAITING FOR BUFFER
4382  015356  001011                        BNE     59$             ;YES, ENQUEUE COMMAND IN BUFFER WAIT QUEUE
4383  015360  004037  022616                JSR     R0,BUFFAL       ;GO ALLOCATE BUFFER
4384  015364  015402                        59$                     ;NOT ENOUGH ROOM, ENQUEUE IN BUFFER WAIT QUEUE
4385  015366  005037  177776                CLR     PS              ;ALLOW RK06 INTERRUPTS
4386  015372  004737  023234                JSR     PC,LDDATA       ;LOAD BUFFER
```

```
 4387  015376  000137  014040                    JMP     ASN3$           ;GO ISSUE COMMAND
 4388
 4389  015402  004037  051564          59$:      JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN BUFFER
 4390  015406  001320                             BWAITQ                  ;  WAIT QUEUE
 4391  015410  000207                    60$:     RTS     PC              ;RETURN
 4392
 4393  015412  132765  000020  000210  WRTVER:  BITB    #BIT4,P.ASSN(R5) ;CHECK IF WRITE PACK IN PROGRESS
 4394  015420  001001                             BNE     1$              ;YES, CONTINUE
 4395  015422  000000                             HALT                    ; **** PROGRAM PROBLEM
 4396  015424  032765  000200  000014  1$:      BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK ISSUED
```

/

```
4397  015432  001410                          BEQ     WV0$            ;YES, CHECK FOR DROP DRIVE
4398  015434  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;CLEAR ISSUE WRITE BEFORE WRITE CHECK
4399  015442  112765  000131  000001          MOVB    #WRTCHK,P.CMND(R5)      ;RESTORE WRITE CHK CMD
4400  015450  000137  014040                  JMP     ASN3$           ;GO ISSUE WRITE CHECK
4401
4402  015454  032765  002000  000014  WV0$:   BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
4403  015462  001402                          BEQ     10$             ;NO, CONTINUE
4404  015464  000137  016206                  JMP     30$             ;YES, TERMINATE PACK WRITING
4405
4406  015470  132765  000040  000210  10$:   BITB    #BIT5,P.ASSN(R5) ;CHECK IF END OF PACK
4407  015476  001402                          BEQ     15$             ;NO, CONTINUE
4408  015500  000137  016230                  JMP     WV2$            ;YES, RETURN BUFFER
4409  015504  052765  000200  000014  15$:   BIS     #W.WCK,P.PRST(R5)       ;SET ISSUE WRITE BEFORE WRT CHK
4410  015512  122765  000140  000001          CMPB    #RELEAS,P.CMND(R5)      ;SEE IF LAST CMD WAS RELEASE
4411  015520  001452                          BEQ     3$                      ;BR IF YES,ELSE FILL PARAM BLOCK
4412  015522  112737  177777  001506          MOVB    #-1,I.ISRL      ;INDICATE THAT AN INTERRUPT HAS BEEN ISSUED
4413  015530  012762  000300  000000          MOV     #INTR,RKCS1(R2) ;GO SCAN FOR DRIVE INTERRUPTS
4414  015536  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4415  015542  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
4416  015550  016565  000124  000136          MOV     P.RCYL(R5),P.LCYL(R5) ;STORE PREVIOUS CYLINDER
4417  015556  016565  000126  000140          MOV     P.RSEC(R5),P.LSEC(R5) ;STORE PREVIOUS TRACK AND SECTOR
4418  015564  116565  000122  000144          MOVB    P.RDPT(R5),P.LDPT(R5) ;STORE PREVIOUS DATA PATTERN
4419  015572  016565  000130  000142          MOV     P.RWC(R5),P.LWC(R5) ;STORE PREVIOUS WORD COUNT
4420  015600  016565  000026  000004          MOV     P.DTS(R5),P.SECT(R5) ;LOAD NEW SECTOR AND TRACK
4421  015606  016565  000026  000126          MOV     P.DTS(R5),P.RSEC(R5)
4422  015614  016565  000030  000002          MOV     P.DCYL(R5),P.CYLN(R5) ;LOAD NEW CYLINDER
4423  015622  016565  000030  000124          MOV     P.DCYL(R5),P.RCYL(R5)
4424  015630  016565  000132  000010          MOV     P.RBAL(R5),P.BALO(R5) ;REINITIALIZE BUS ADDRESS FOR POSSIBLE
4425                                                  ;  BAD SECTOR RECOVERY ON LAST COMMAND
4426  015636  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)      ;SEE IF LAST CMD WAS WRT CHK
4427  015644  001404                          BEQ     4$              ;BR IF YES
4428  015646  112765  000131  000001  3$:     MOVB    #WRTCHK,P.CMND(R5)      ;ELSE SET TO DO WRT DATA (W.WCK=1)
4429  015654  000410                          BR      5$
4430  015656  042765  000200  000014  4$:     BIC     #W.WCK,P.PRST(R5)
4431  015664  112765  000140  000001          MOVB    #RELEAS,P.CMND(R5)      ;RELEAS AFTER WRT CHK
4432  015672  000137  014040                  JMP     ASN3$           ;GO RELEASE
4433  015676  016546  000002          5$:     MOV     P.CYLN(R5),-(SP) ;STORE CYLINDER ADDRESS ON STACK
4434  015702  006316                          ASL     (SP)            ;MULTIPLY CYLINDER BY 3
4435  015704  066516  000002                  ADD     P.CYLN(R5),(SP)
4436  015710  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK FOR TRACK
4437  015712  116516  000005                  MOVB    P.TRCK(R5),(SP) ;STORE TRACK ON STACK
4438  015716  061666  000002                  ADD     (SP),2(SP)      ;ADD TRACK
4439  015722  012716  000026                  MOV     #22.,(SP)       ;STORE 22 ON STACK (MULTIPLIER)
4440  015726  004737  055210                  JSR     PC,$MULT        ;CALCULATE NUMBER OF SECTORS TRANSFERRED
4441  015732  012616                          MOV     (SP)+,(SP)      ;THROW AWAY MOST SIGNIFICANT BITS
4442  015734  005046                          CLR     -(SP)           ;CLEAR LOCATION FOR SECTOR
4443  015736  116516  000004                  MOVB    P.SECT(R5),(SP) ;STORE SECTOR COUNT
4444  015742  061666  000002                  ADD     (SP),2(SP)      ;DETERMINE TOTAL NUMBER OF SECTORS TRANSFERRED
4445  015746  004737  027252                  JSR     PC,GETNUM       ;GET CORRECT LAST CYL NUM
4446  015752  013716  027336                  MOV     HOLD,(SP)       ;STORE NUMBER OF SECTORS ON STACK
4447  015756  166616  000002                  SUB     2(SP),(SP)      ;DETERMINE NUMBER OF SECTORS LEFT ON
4448                                                                  ;  ON PACK TO BE WRITTEN
4449  015762  016546  000130                  MOV     P.RWC(R5),-(SP) ;STORE WORD COUNT
4450  015766  011665  000012                  MOV     (SP),P.WC(R5)
4451  015772  005416                          NEG     (SP)            ;MAKE IT POSITIVE
4452  015774  116616  000001                  MOVB    1(SP),(SP)      ;KEEP SECTOR COUNT
```

```
 4453  016000  105066  000001                    CLRB    1(SP)              ;CLEAR MOST SIGNIFICANT BITS
 4454  016004  021666  000002                    CMP     (SP),2(SP)         ;CHECK IF PACK OVERRUN
 4455  016010  103413                           BLO     20$                ;NO, GO ISSUE COMMAND
 4456  016012  001407                           BEQ     19$                ;CHECK IF NO NEW WORD COUNT NEEDED
 4457  016014  005016                           CLR     (SP)               ;MAKE ROOM FOR NEW WORD COUNT
 4458  016016  116666  000002  000001           MOVB    2(SP),1(SP)        ;MOVE IN NUMBER OF SECTORS LEFT
 4459  016024  005416                           NEG     (SP)               ;MAKE WORD COUNT NEGATIVE
 4460  016026  011665  000012                    MOV     (SP),P.WC(R5)      ;LOAD NEW WORD COUNT
 4461  016032  152765  000040  000210  19$:     BISB    #BIT5,P.ASSN(R5)   ;SET END OF PACK TRANSFER
 4462
 4463  016040  062706  000006          20$:     ADD     #6,SP              ;ADJUST STACK
 4464  016044  016546  000012                    MOV     P.WC(R5),-(SP)     ;STORE PRESENT WORD COUNT
 4465  016050  005416                           NEG     (SP)               ;MAKE IT POSITIVE
 4466  016052  062765  000002  000152           ADD     #2,P.NODR(R5)      ;INCREMENT NUMBER OF ORDERS
 4467  016060  061665  000156                    ADD     (SP),P.NWRT(R5)    ;ADD NUMBER OF WORDS WRITTEN
 4468  016064  005565  000160                    ADC     P.NWRT+2(R5)
 4469  016070  062665  000164                    ADD     (SP)+,P.NRD(R5)    ;ADD NUMBER OF WORDS READ
 4470  016074  005565  000166                    ADC     P.NRD+2(R5)
 4471  016100  105265  000122                    INCB    P.RDPT(R5)         ;GENERATE NEXT DATA PATTERN TO BE WRITTEN
 4472  016104  142765  000360  000122           BICB    #360,P.RDPT(R5)    ;KEEP 4 LEAST SIGNIFICANT BITS
 4473  016112  116565  000122  000121           MOVB    P.RDPT(R5),P.DPAT(R5) ;STORE DATA PATTERN USED
 4474  016120  013737  001452  177776           MOV     RKPRI,PS           ;LOCK OUT RK06 INTERRUPTS
 4475  016126  132765  000040  000210           BITB    #BIT5,P.ASSN(R5)   ;CHECK IF WRITE TO END OF PACK
 4476  016134  001014                           BNE     25$                ;YES, RELEASE BUFFER
 4477  016136  005737  001320                    TST     BWAITQ             ;CHECK IF BUFFER WAIT QUEUE EMPTY
 4478  016142  001011                           BNE     25$                ;NO, RELEASE BUFFER
 4479  016144  016565  000012  000130           MOV     P.WC(R5),P.RWC(R5) ;STORE CURRENT GENERATED WORD COUNT
 4480  016152  005037  177776                    CLR     PS                 ;ALLOW RK06 INTERRUPTS
 4481  016156  004737  023234                    JSR     PC,LDDATA          ;LOAD NEW RANDOM DATA
 4482  016162  000137  014040                    JMP     ASN3$              ;GO ISSUE COMMAND
 4483
 4484  016166  004737  023020          25$:     JSR     PC,BUFREL          ;RELEASE BUFFER FOR OTHER DRIVES
 4485  016172  004037  051564                    JSR     R0,Q.PUSH          ;PUT PARAMETER BLOCK ON BUFFER WAIT QUEUE
 4486  016176  001320                           BWAITQ
 4487  016200  004737  022520                    JSR     PC,GETBUF          ;GET NEW BUFFER
 4488  016204  000207                           RTS     PC                 ;RETURN
 4489
 4490  016206  105065  000210          30$:     CLRB    P.ASSN(R5)         ;CLEAR ASSIGNMENT CODE
 4491  016212  004737  023020                    JSR     PC,BUFREL          ;RELEASE BUFFER
 4492  016216  104401  056656          WV1$:    TYPE    ,OPR008            ;TYPE 'OPERATOR INITIATED DROP DRIVE''
 4493  016222  004737  013162                    JSR     PC,DROP            ;GO DROP DRIVE
 4494  016226  000444                           BR      WV5$               ;GO ALLOCATE BUFFER
 4495
 4496  016230  105065  000210          Wv2$:    CLRB    P.ASSN(R5)         ;CLEAR ASSIGNMENT FLAGS
 4497  016234  116537  000000  056755           MOVB    P.DRVN(R5),OPR011  ;GET DRIVE NUMBER
 4498  016242  152737  000060  056755           BISB    #60,OPR011         ;MAKE IT ASCII
 4499  016250  104401  056736                    TYPE    ,OPR010            ;TYPE 'DRIVE NUMBER N UNDER TEST''
 4500  016254  104401  056501                    TYPE    ,OPR003
 4501  016260  004737  027774                    JSR     PC,PRISER          ;PRINT DRIVE AND PACK SERIAL NUMBERS
 4502  016264  004737  044046                    JSR     PC,PRITIM          ;PRINT TIME
 4503  016270  104401  001165                    TYPE    ,$CRLF
 4504  016274  010346                           MOV     R3,-(SP)           ;STORE R3 ON STACK
 4505  016276  010446                           MOV     R4,-(SP)           ;STORE R4 ON STACK
 4506  016300  010503                           MOV     R5,R3              ;LOAD PARAMETER BLOCK BASE
 4507  016302  062703  000152                    ADD     #P.NODR,R3         ;CALCULATE BEGINNING OF STATISTICS
 4508  016306  010504                           MOV     R5,R4              ;LOAD PARAMETER BLOCK BASE
```

```
 4509  016310  062704  000210                ADD     #P.ASSN,R4      ;CALCULATE END OF STATISTICS
 4510  016314  005023                7$:     CLR     (R3)+           ;CLEAR STATISTICS BEFORE STARTING TEST
 4511  016316  020304                        CMP     R3,R4           ;CHECK IF FINISHED
 4512  016320  001375                        BNE     7$              ;NO, CONTINUE
 4513  016322  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4514  016324  012603                        MOV     (SP)+,R3        ;RESTORE R3
 4515  016326  004737  023020                JSR     PC,BUFREL       ;RELEASE BUFFER
 4516  016332  004037  051564                JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN DRIVE
 4517  016336  001314                        AVAILQ                  ;  AVAILIABLE QUEUE
 4518  016340  004737  022520        WV5$:   JSR     PC,GETBUF       ;GET NEW BUFFER
 4519  016344  000207                WV9$:   RTS     PC              ;RETURN
```

```
                                        .SBTTL  ALTER DRIVE PARAMETERS

                                PARM:   MOV     R4,-(SP)        ;STORE R4 ON STACK
                                        MOV     R3,-(SP)        ;STORE R3 ON STACK
060131                          P0$:    MOVB    P.DRVN(R5),PAR001 ;LOAD DRIVE NUMBER FOR PRINT OUT
060131                                  BISB    #60,PAR00       ;MAKE IT ASCII
                                        TYPE    ,PAR000         ;TYPE "PARAMETERS FOR DRIVE N"
                                        JSR     PC,DTYPE        ;INPUT DRIVE TYPE
                                2$:     TYPE    ,PAR002         ;TYPE "CYLINDER MAX,MIN"
                                        JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
                                        5$                      ;COMMA DETECTED
                                        6$                      ;CARRIAGE RETURN DETECTED
                                        P1$                     ;CONTROL Z <^Z> DETECTED
                                        P0$                     ;CONTROL C <^C> DETECTED
                                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
                                        JSR     PC,OCTBIN       ;CONVERT TO BINARY
                                        4$                      ;ERROR RETURN
001602                                  MOV     (SP)+,PARCYL    ;STORE MAX CYLINDER ENTRY
027252                                  JSR     PC,GETNUM       ;GET CORRECT LAST CYL NUM
027332  001602                          CMP     LSTCYL,PARCYL   ;CHECK IF LESS OR EQUAL TO
                                                                ;   MAXIMUM CYLINDER ADDRESS
                                        BLO     4$              ;NO, TRY AGAIN
                                        MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
                                64$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
                                        BEQ     7$              ;YES, EXIT
000054                                  CMPB    (R4)+,#',       ;CHECK FOR COMMA
                                        BNE     64$             ;NO, GO TEST NEXT CHARACTER
                                3$:     MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
                                        JSR     PC,OCTBIN       ;CONVERT TO BINARY
                                        4$                      ;ERROR RETURN
001604                                  MOV     (SP)+,PARCYL+2  ;STORE MIN CYLINDER ENTRY
001602  001604                          CMP     PARCYL,PARCYL+2 ;CHECK IF LEGAL ENTRY
                                        BHIS    8$              ;YES, LOAD MAX AND MIN CYLINDER
                                4$:
                                        MOV     R3,65$          ;LOAD BUFFER ADDRESS
                                        TYPE                    ;TYPE RECEIVED INPUT
                                65$:    .WORD   0               ;BUFFER ADDRESS
001164                                  TYPE    ,$QUES          ;TYPE QUESTION MARK
                                        BR      2$              ;TRY AGAIN

001602                          5$:     MOV     LSTCYL,PARCYL   ;LOAD MAX CYLINDER = 632/1456
                                        MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
                                        TSTB    (R4)+           ;ADJUST R4
                                        BR      3$              ;GET MINIMUM CYLINDER

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

060161                          10$:    TYPE    ,PAR003         ;TYPE "TRACK MAX,MIN"
020516                                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
                                        13$                     ;COMMA DETECTED
                                        14$                     ;CARRIAGE RETURN DETECTED
                                        P2$                     ;CONTROL Z <^Z> DETECTED
                                        P0$                     ;CONTROL C <^C> DETECTED
```

```
4576  016606  010346                        MOV     R3,-(SP)          ;STORE BUFFER ADDRESS
4577  016610  004737  054406                JSR     PC,OCTBIN         ;CONVERT TO BINARY
4578  016614  016672                        12$                       ;ERROR RETURN
4579  016616  012637  001606                MOV     (SP)+,PARTRK      ;STORE MAX TRACK ENTRY
4580  016622  022737  000002  001606        CMP     #D.MXTR,PARTRK    ;CHECK IF LESS OR EQUAL TO
4581                                                                  ;  MAXIMUM TRACK ADDRESS
4582  016630  103420                        BLO     12$               ;NO, TRY AGAIN
4583  016632  010304                        MOV     R3,R4             ;STORE R3 IN R4 FOR LINE SCAN
4584  016634  105714                66$:    TSTB    (R4)              ;CHECK FOR CARRIAGE RETURN
4585  016636  001435                        BEQ     15$               ;YES, EXIT
4586  016640  122427  000054                CMPB    (R4)+,#',         ;CHECK FOR COMMA
4587  016644  001373                        BNE     66$               ;NO, GO TEST NEXT CHARACTER
4588  016646  010446                11$:    MOV     R4,-(SP)          ;STORE BUFFER ADDRESS
4589  016650  004737  054406                JSR     PC,OCTBIN         ;CONVERT TO BINARY
4590  016654  016672                        12$                       ;ERROR RETURN
4591  016656  012637  001610                MOV     (SP)+,PARTRK+2    ;STORE MIN TRACK ENTRY
4592  016662  023737  001606  001610        CMP     PARTRK,PARTRK+2   ;CHECK IF LEGAL ENTRY
4593  016670  103022                        BHIS    16$               ;YES, LOAD MAX AND MIN TRACK
4594  016672                        12$:
4595  016672  010337  016700                MOV     R3,67$            :LOAD BUFFER ADDRESS
4596  016676  104401                        TYPE                      ;TYPE RECEIVED INPUT
4597  016700  000000                67$:    .WORD   0                 ;BUFFER ADDRESS
4598  016702  104401  001164                TYPE    ,$QUES            ;TYPE QUESTION MARK
4599  016706  000727                        BR      10$               ;TRY AGAIN
4600
4601  016710  012737  000002  001606  13$:  MOV     #D.MXTR,PARTRK    ;LOAD MAX TRACK = 2
4602  016716  010304                        MOV     R3,R4             ;STORE R3 IN R4 FOR LINE SCAN
4603  016720  105724                        TSTB    (R4)+             ;ADJUST R4
4604  016722  000751                        BR      11$               ;GET MINIMUM TRACK
4605
4606  016724  012737  000002  001606  14$:  MOV     #D.MXTR,PARTRK    ;LOAD MAX TRACK = 2
4607  016732  005037  001610        15$:    CLR     PARTRK+2          ;LOAD MIN TRACK = 0
4608  016736  113765  001606  000073  16$:  MOVB    PARTRK,P.MXTR(R5) ;LOAD MAX TRACK
4609  016744  113765  001610  000072        MOVB    PARTRK+2,P.MNTR(R5) ;LOAD MIN TRACK
4610
4611  016752  104401  060201        18$:    TYPE    ,PAR004           ;TYPE "SECTOR MAX,MIN"
4612  016756  004037  020516                JSR     R0,TSTDEF         ;CHECK FOR DEFAULT PARAMETERS
4613  016762  017074                        21$                       ;COMMA DETECTED
4614  016764  017110                        22$                       ;CARRIAGE RETURN DETECTED
4615  016766  020402                        P3$                       ;CONTROL Z <^Z> DETECTED
4616  016770  016352                        P0$                       ;CONTROL C <^C> DETECTED
4617  016772  010346                        MOV     R3,-(SP)          ;STORE BUFFER ADDRESS
4618  016774  004737  054406                JSR     PC,OCTBIN         ;CONVERT TO BINARY
4619  017000  017056                        20$                       ;ERROR RETURN
4620  017002  012637  001612                MOV     (SP)+,PARSEC      ;STORE MAX SECTOR ENTRY
4621  017006  022737  000025  001612        CMP     #D.MXSC,PARSEC    ;CHECK IF LESS OR EQUAL TO
4622                                                                  ;  MAXIMUM SECTOR ADDRESS
4623  017014  103420                        BLO     20$               ;NO, TRY AGAIN
4624  017016  010304                        MOV     R3,R4             ;STORE R3 IN R4 FOR LINE SCAN
4625  017020  105714                68$:    TSTB    (R4)              ;CHECK FOR CARRIAGE RETURN
4626  017022  001435                        BEQ     23$               ;YES, EXIT
4627  017024  122427  000054                CMPB    (R4)+,#',         ;CHECK FOR COMMA
4628  017030  001373                        BNE     68$               ;NO, GO TEST NEXT CHARACTER
4629  017032  010446                19$:    MOV     R4,-(SP)          ;STORE BUFFER ADDRESS
4630  017034  004737  054406                JSR     PC,OCTBIN         ;CONVERT TO BINARY
4631  017040  017056                        20$                       ;ERROR RETURN
```

```
4632  017042  012637  001614              MOV     (SP)+,PARSEC+2  ;STORE MIN SECTOR ENTRY
4633  017046  023737  001612  001614      CMP     PARSEC,PARSEC+2 ;CHECK IF LEGAL ENTRY
4634  017054  103022                      BHIS    24$             ;YES, LOAD MAX AND MIN SECTOR
4635  017056                      20$:
4636  017056  010337  017064              MOV     R3,69$          ;LOAD BUFFER ADDRESS
4637  017062  104401                      TYPE                    ;TYPE RECEIVED INPUT
4638  017064  000000              69$:    .WORD   0               ;BUFFER ADDRESS
4639  017066  104401  001164              TYPE    ,$QUES          ;TYPE QUESTION MARK
4640  017072  000727                      BR      18$             ;TRY AGAIN
4641
4642  017074  012737  000025  001612  21$:MOV     #D.MXSC,PARSEC  ;LOAD MAX SECTOR = 21
4643  017102  010304                      MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
4644  017104  105724                      TSTB    (R4)+           ;ADJUST R4
4645  017106  000751                      BR      19$             ;GET MINIMUM SECTOR
4646
4647  017110  012737  000025  001612  22$:MOV     #D.MXSC,PARSEC  ;LOAD MAX SECTOR = 21
4648  017116  005037  001614      23$:CLR     PARSEC+2        ;LOAD MIN SECTOR = 0
4649  017122  113765  001612  000075  24$:MOVB    PARSEC,P.MXSC(R5) ;LOAD MAX SECTOR
4650  017130  113765  001614  000074      MOVB    PARSEC+2,P.MNSC(R5) ;LOAD MIN SECTOR
4651
4652  017136  104401  060222          26$:TYPE    ,PAR005         ;TYPE 'READ/WRITE RATIO ='
4653  017142  004037  020516              JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4654  017146  017176                      27$                     ;COMMA DETECTED
4655  017150  017222                      29$                     ;CARRIAGE RETURN DETECTED
4656  017152  020414                      P4$                     ;CONTROL Z <^Z> DETECTED
4657  017154  016352                      P0$                     ;CONTROL C <^C> DETECTED
4658  017156  010346                      MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
4659  017160  004737  054406              JSR     PC,OCTBIN       ;CONVERT BINARY
4660  017164  017176                      27$                     ;ERROR RETURN
4661  017166  012604                      MOV     (SP)+,R4        ;STORE INPUT
4662  017170  022704  000010              CMP     #10,R4          ;CHECK IF 0-7
4663  017174  101007                      BHI     28$             ;YES, LOAD READ/WRITE RATIO
4664  017176                      27$:
4665  017176  010337  017204              MOV     R3,70$          ;LOAD BUFFER ADDRESS
4666  017202  104401                      TYPE                    ;TYPE RECEIVED INPUT
4667  017204  000000              70$:    .WORD   0               ;BUFFER ADDRESS
4668  017206  104401  001164              TYPE    ,$QUES          ;TYPE QUESTION MARK
4669  017212  000751                      BR      26$             ;TRY AGAIN
4670
4671  017214  110465  000076          28$:MOVB    R4,P.RATE(R5)   ;LOAD READ/WRITE RATIO
4672  017220  000403                      BR      30$             ;ASK 'WRITE CHECK AFTER WRITE ='
4673
4674  017222  112765  000003  000076  29$:MOVB    #D.RATE,P.RATE(R5) ;LOAD DEFAULT READ/WRITE
4675
4676  017230  104401  060245          30$:TYPE    ,PAR006         ;TYPE 'WRITE CHECK AFTER WRITE ='
4677  017234  004037  020516              JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4678  017240  017266                      31$                     ;COMMA DETECTED
4679  017242  017304                      32$                     ;CARRIAGE RETURN DETECTED
4680  017244  020422                      P5$                     ;CONTROL Z <^Z> DETECTED
4681  017246  016352                      P0$                     ;CONTROL C <^C> DETECTED
4682  017250  122713  000131              CMPB    #'Y,(R3)        ;CHECK IF YES
4683  017254  001013                      BNE     32$             ;NO, LOAD DEFAULT
4684  017256  112765  177777  000077      MOVB    #-1,P.AWCK(R5)  ;SET WRITE CHECK AFTER WRITE
4685  017264  000411                      BR      34$             ;GET CORRECTABLE READ ERROR
4686                                                              ;  THRESHOLD
4687
```

```
                                31$:
                                        MOV     R3,71$          ;LOAD BUFFER ADDRESS
                                        TYPE                    ;TYPE RECEIVED INPUT
                                71$:    .WORD   0               ;BUFFER ADDRESS
                                        TYPE    ,$QUES          ;TYPE QUESTION MARK
                                        BR      30$             ;TRY AGAIN

                                32$:    CLRB    P.AWCK(R5)      ;CLEAR WRITE CHECK AFTER WRITE

                                34$:    TYPE    ,PAR007         ;TYPE ''CORRECTABLE READ ERROR
                                                                ;  THRESHOLD
                                        JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
                                        35$                     ;COMMA DETECTED
                                        36$                     ;CARRIAGE RETURN DETECTED
                                        P6$                     ;CONTROL Z <^Z> DETECTED
                                        P0$                     ;CONTROL C <^C> DETECTED
                                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
                                        JSR     PC,DECBIN       ;CONVERT TO BINARY
                                        35$                     ;ERROR RETURN
                                        MOV     (SP)+,P.CERT(R5) ;LOAD CORRECTABLE READ ERROR
                                                                ;  THRESHOLD
                                        BPL     38$             ;IF NOT NEGATIVE, GET UNCORRECTABLE
                                                                ;  READ ERROR THRESHOLD

                                35$:
                                        MOV     R3,72$          ;LOAD BUFFER ADDRESS
                                        TYPE                    ;TYPE RECEIVED INPUT
                                72$:    .WORD   0               ;BUFFER ADDRESS
                                        TYPE    ,$QUES          ;TYPE QUESTION MARK
                                        BR      34$             ;TRY AGAIN

                                36$:    MOV     #D.CERT,P.CERT(R5) ;LOAD DEFAULT CORRECTABLE READ
                                                                ;  ERROR THRESHOLD

                                38$:    TYPE    ,PAR008         ;TYPE 'UNCORRECTABLE READ ERROR
                                                                ;  THRESHOLD =''
                                        JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
                                        39$                     ;COMMA DETECTED
                                        40$                     ;CARRIAGE RETURN DETECTED
                                        P7$                     ;CONTROL Z <^Z> DETECTED
                                        P0$                     ;CONTROL C <^C> DETECTED
                                        MOV     R3,-(SP)        ;STORE PUFFER ADDRESS
                                        JSR     PC,DECBIN       ;CONVERT TO BINARY
                                        39$                     ;ERROR RETURN
                                        MOV     (SP)+,P.UERT(R5) ;LOAD UNCORRECTABLE READ ERROR
                                                                ;  THRESHOLD
                                        BPL     42$             ;IF NOT NEGATIVE, GET SEEK
                                                                ;  ERROR THRESHOLD

                                39$:
                                        MOV     R3,73$          ;LOAD BUFFER ADDRESS
                                        TYPE                    ;TYPE RECEIVED INPUT
                                73$:    .WORD   0               ;BUFFER ADDRESS
                                        TYPE    ,$QUES          ;TYPE QUESTION MARK
                                        BR      38$             ;TRY AGAIN

                                40$:    MOV     #D.UERT,P.UERT(R5) ;LOAD DEFAULT UNCORRECTABLE READ
```

```
                                                        ;  ERROR THRESHOLD
                                                        ;
017454  104401  060407          42$:    TYPE    ,PAR009         ;TYPE ''SEEK ERROR THRESHOLD =''
017460  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017464  017512                          43$                     ;COMMA DETECTED
017466  017530                          44$                     ;CARRIAGE RETURN DETECTED
017470  020442                          P8$                     ;CONTROL Z <^Z> DETECTED
017472  016352                          P0$                     ;CONTROL C <^C> DETECTED
017474  010346                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
017476  004737  054542                  JSR     PC,DECBIN       ;CONVERT TO BINARY
017502  017512                          43$                     ;ERROR RETURN
017504  012665  000104                  MOV     (SP)+,P.SKET(R5) ;LOAD SEEK ERROR THRESHOLD
017510  100012                          BPL     46$             ;CHECK IF INHIBIT ERROR CORRECTION
                                                        ;  AND RETRY
017512                          43$:
017512  010337  017520                  MOV     R3,74$          ;LOAD BUFFER ADDRESS
017516  104401                          TYPE                    ;TYPE RECEIVED INPUT
017520  000000                  74$:    .WORD   0               ;BUFFER ADDRESS
017522  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
017526  000752                          BR      42$             ;TRY AGAIN
                                                        ;
017530  012765  000005  000104  44$:    MOV     #D.SKET,P.SKET(R5) ;LOAD DEFAULT SEEK ERROR THRESHOLD
                                                        ;
017536  104401  060436          46$:    TYPE    ,PAR010         ;TYPE ''INHIBIT ERROR CORRECTION
                                                        ;  AND RETRY
017542  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017546  017574                          47$                     ;COMMA DETECTED
017550  017612                          48$                     ;CARRIAGE RETURN DETECTED
017552  020450                          P9$                     ;CONTROL Z <^Z> DETECTED
017554  016352                          P0$                     ;CONTROL C <^C> DETECTED
017556  122713  000131                  CMPB    #'Y,(R3)        ;CHECK IF YES
017562  001013                          BNE     48$             ;LOAD DEFAULT
017564  112765  177777  000120          MOVB    #-1,P.IERC(R5)  ;SET INHIBIT ERROR CORRECTION
017572  000411                          BR      50$             ;GET COMMAND COUNT THRESHOLD
                                                        ;
017574                          47$:
017574  010337  017602                  MOV     R3,75$          ;LOAD BUFFER ADDRESS
017600  104401                          TYPE                    ;TYPE RECEIVED INPUT
017602  000000                  75$:    .WORD   0               ;BUFFER ADDRESS
017604  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
017610  000752                          BR      46$             ;TRY AGAIN
                                                        ;
017612  105065  000120          48$:    CLRB    P.IERC(R5)      ;CLEAR INHIBIT ERROR CORRECTION
                                                        ;
017616  104401  060503          50$:    TYPE    ,PAR011         ;TYPE ''OPERATION COUNT THRESHOLD*65K =''
017622  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017626  017672                          51$                     ;COMMA DETECTED
017630  017710                          52$                     ;CARRIAGE RETURN DETECTED
017632  020454                          P10$                    ;CONTROL Z <^Z> DETECTED
017634  016352                          P0$                     ;CONTROL C <^C> DETECTED
017636  010346                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
017640  004737  054542                  JSR     PC,DECBIN       ;CONVERT TO BINARY
017644  017672                          51$                     ;ERROR RETURN
017646  012665  000110                  MOV     (SP)+,P.MXCD+2(R5) ;STORE NUMBER OF 65K BLOCKS
017652  001407                          BEQ     51$             ;IF ZERO, TRY AGAIN
017654  100406                          BMI     51$             ;IF MINUS, TRY AGAIN
```

```
 4800  017656  005365  000110                DEC     P.MXCD+2(R5)    ;DECREMENT BY 1
 4801  017662  012765  177777  000106        MOV     #-1,P.MXCD(R5)
 4802  017670  000415                        BR      54$             ;GET WORD TRANSFERREC THRESHOLD
 4803
 4804  017672                        51$:
 4805  017672  010337  017700                MOV     R3,76$          ;LOAD BUFFER ADDRESS
 4806  017676  104401                        TYPE                    ;TYPE RECEIVED INPUT
 4807  017700  000000                76$:    .WORD   0               ;BUFFER ADDRESS
 4808  017702  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
 4809  017706  000743                        BR      50$             ;TRY AGAIN
 4810
 4811  017710  012765  077777  000110  52$:  MOV     #D.CTRH,P.MXCD+2(R5) ;LOAD DEFAULT COMMAND THRESHOLD
 4812  017716  012765  177770  000106        MOV     #D.CTRL,P.MXCD(R5)
 4813
 4814  017724  104401  060543          54$:  TYPE    ,PAR012         ;TYPE 'WORD TRNSFERRED THRESHOLD*520K =''
 4815  017730  004037  020516                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
 4816  017734  020032                        55$                     ;COMMA DETECTED
 4817  017736  020050                        56$                     ;CARRIAGE RETURN DETECTED
 4818  017740  020470                        P11$                    ;CONTROL Z <^Z> DETECTED
 4819  017742  016352                        P0$                     ;CONTROL C <^C> DETECTED
 4820  017744  010346                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
 4821  017746  004737  054542                JSR     PC,DECBIN       ;CONVERT TO BINARY
 4822  017752  020032                        55$                     ;ERROR RETURN
 4823  017754  012665  000112                MOV     (SP)+,P.MXWT(R5) ;STORE NUMBER OF 520K BLOCKS
 4824  017760  001424                        BEQ     55$             ;IF ZERO, TRY AGAIN
 4825  017762  100423                        BMI     55$             ;IF NEGATIVE, TRY AGAIN
 4826  017764  005065  000114                CLR     P.MXWT+2(R5)    ;CLEAR HIGH ORDER BITS
 4827  017770  006165  000112                ROL     P.MXWT(R5)      ;SHIFT 4 BITS LEFT
 4828  017774  006165  000114                ROL     P.MXWT+2(R5)
 4829  020000  006165  000112                ROL     P.MXWT(R5)
 4830  020004  006165  000114                ROL     P.MXWT+2(R5)
 4831  020010  006165  000112                ROL     P.MXWT(R5)
 4832  020014  006165  000114                ROL     P.MXWT+2(R5)
 4833  020020  006165  000112                ROL     P.MXWT(R5)
 4834  020024  006165  000114                ROL     P.MXWT+2(R5)
 4835  020030  000415                        BR      60$             ;CHECK FOR SAMPLED COMPARES
 4836
 4837  020032                        55$:
 4838  020032  010337  020040                MOV     R3,77$          ;LOAD BUFFER ADDRESS
 4839  020036  104401                        TYPE                    ;TYPE RECEIVED INPUT
 4840  020040  000000                77$:    .WORD   0               ;BUFFER ADDRESS
 4841  020042  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
 4842  020046  000726                        BR      54$             ;TRY AGAIN
 4843
 4844  020050  012765  077777  000114  56$:  MOV     #D.WTHI,P.MXWT+2(R5) ;LOAD DEFAULT WORD TRANSFER
 4845  020056  012765  177770  000112        MOV     #D.WTLO,P.MXWT(R5)      ;  THRESHOLD
 4846
 4847  020064  005737  001362          60$:  TST     SOFCMF          ;CHECK IF ANY SOFTWARE COMPARES ARE TO BE MADE
 4848  020070  001430                        BEQ     PREXAR          ;NO, DETERMINE TRACK EXCLUSION
 4849  020072  104401  061041                TYPE    ,PAR021         ;TYPE 'SAMPLED COMPARES =''
 4850  020076  004037  020516                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
 4851  020102  020130                        61$                     ;COMMA DETECTED
 4852  020104  020146                        62$                     ;CARRIAGE RETURN DETECTED
 4853  020106  020504                        P12$                    ;CONTROL Z <^Z> DETECTED
 4854  020110  016352                        P0$                     ;CONTROL C <^C> DETECTED
 4855  020112  122713  000116                CMPB    #'N,(R3)        ;CHECK IF NO
```

```
4856  020116  001013                    BNE     62$             ;NO, MAKE SAMPLED COMPARES
4857  020120  112765  177777  000116    MOVB    #-1,P.SMPL(R5)  ;DO NOT MAKE SAMPLED COMPARES
4858  020126  000411                    BR      PREXAR          ;DETERMINE TRACK EXCLUSION
4859
4860  020130                    61$:
4861  020130  010337  020136            MOV     R3,78$          ;LOAD BUFFER ADDRESS
4862  020134  104401                    TYPE                    ;TYPE RECEIVED INPUT
4863  020136  000000            78$:    .WORD   0               ;BUFFER ADDRESS
4864  020140  104401  001164            TYPE    ,$QUES          ;TYPE QUESTION MARK
4865  020144  000747                    BR      60$             ;TRY AGAIN
4866
4867  020146  105065  000116    62$:    CLRB    P.SMPL(R5)      ;MAKE SAMPLED COMPARES
4868
4869  020152  104401  060676    PREXAR: TYPE    ,PAR019         ;TYPE "CHANGE EXCUDED PACK AREA ="
4870  020156  104401  060650            TYPE    ,PAR018
4871  020162  004037  020516            JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4872  020166  020210                    1$                      ;COMMA DETECTED
4873  020170  020350                    50$                     ;CARRIAGE RETURN DETECTED
4874  020172  020350                    50$                     ;CONTROL Z <^Z> DETECTED
4875  020174  016352                    P0$                     ;CONTROL C <^C> DETECTED
4876  020176  122713  000131            CMPB    #'Y,(R3)        ;CHECK IF YES
4877  020202  001411                    BEQ     2$              ;YES, LOAD NEW TRACK EXCULSIONS
4878  020204  000137  020350            JMP     50$             ;NO, RETURN
4879
4880  020210                    1$:
4881  020210  010337  020216            MOV     R3,64$          ;LOAD BUFFER ADDRESS
4882  020214  104401                    TYPE                    ;TYPE RECEIVED INPUT
4883  020216  000000            64$:    .WORD   0               ;BUFFER ADDRESS
4884  020220  104401  001164            TYPE    ,$QUES          ;TYPE QUESTION MARK
4885  020224  000752                    BR      PREXAR          ;TRY AGAIN
4886
4887  020226  010146            2$:     MOV     R1,-(SP)        ;STORE R1 ON STACK
4888  020230  010501                    MOV     R5,R1           ;STORE PARMETER BLOCK ADDRESS
4889  020232  062701  000232            ADD     #P.EXAR,R1      ;CALCULATE EXCLUSION AREA
4890  020236  012704  000012            MOV     #10.,R4         ;LOAD COUNT
4891  020242  005021            5$:     CLR     (R1)+           ;CLEAR EXCLUSION AREA
4892  020244  005304                    DEC     R4              ;DECREMENT COUNT
4893  020246  001375                    BNE     5$              ;CHECK IF DONE
4894  020250  162701  000024            SUB     #20.,R1         ;REINITIALIZE POINTER
4895  020254  004037  020664            JSR     R0,LDEXAR       ;GET FIRST EXCLUDED AREA
4896  020260  060610                    PAR013
4897  020262  020346                    45$                     ;DEFAULT RETURN
4898  020264  020340                    40$                     ;CONTROL C <^C> RETURN
4899  020266  004037  020664            JSR     R0,LDEXAR       ;GET SECOND EXCLUDED AREA
4900  020272  060616                    PAR014
4901  020274  020346                    45$                     ;DEFAULT RETURN
4902  020276  020340                    40$                     ;CONTROL C <^C> RETURN
4903  020300  004037  020664            JSR     R0,LDEXAR       ;GET THIRD EXCLUDED AREA
4904  020304  060625                    PAR015
4905  020306  020346                    45$                     ;DEFAULT RETURN
4906  020310  020340                    40$                     ;CONTROL C <^C> RETURN
4907  020312  004037  020664            JSR     R0,LDEXAR       ;GET FOURTH EXCLUDED AREA
4908  020316  060633                    PAR016
4909  020320  020346                    45$                     ;DEFAULT RETURN
4910  020322  020340                    40$                     ;CONTROL L <^C> RETURN
4911  020324  004037  020664            JSR     R0,LDEXAR       ;GET FIFTH EXCLUDED AREA
```

```
 4912  020330  060642                          PAR017
 4913  020332  020346                          45$                     ;DEFAULT RETURN
 4914  020334  020340                          40$                     ;CONTROL C <^C> RETURN
 4915  020336  000403                          BR      45$             ;RETURN
 4916
 4917  020340  012601                  40$:    MOV     (SP)+,R1        ;RESTORE R1
 4918  020342  000137  016352                  JMP     P0$             ;START AGAIN
 4919
 4920  020346  012601                  45$:    MOV     (SP)+,R1        ;RESTORE R1
 4921  020350  012603                  50$:    MOV     (SP)+,R3        ;RESTORE R3
 4922  020352  012604                          MOV     (SP)+,R4        ;RESTORE R4
 4923  020354  000207                          RTS     PC              ;RETURN
 4924
 4925  020356  013765  027332  000070  P1$:    MOV     LSTCYL,P.MXCL(R5) ;LOAD DEFAULT MAXIMUM CYLINDER = 632/1456
 4926  020364  005065  000066                  CLR     P.MNCL(R5)      ;LOAD DEFAULT MINIMUM CYLINDER
 4927  020370  112765  000002  000073  P2$:    MOVB    #D.MXTR,P.MXTR(R5) ;LOAD DEFAULT MAXIMUM TRACK = 2
 4928  020376  105065  000072                  CLRB    P.MNTR(R5)      ;LOAD DEFAULT MINIMUM TRACK
 4929  020402  112765  000025  000075  P3$:    MOVB    #D.MXSC,P.MXSC(R5) ;LOAD DEFAULT MAXIMUM SECTOR = 21
 4930  020410  105065  000074                  CLRB    P.MNSC(R5)      ;LOAD DEFAULT MINIMUM SECTOR
 4931  020414  112765  000003  000076  P4$:    MOVB    #D.RATE,P.RATE(R5) ;LOAD READ/WRITE RATIO
 4932  020422  105065  000077          P5$:    CLRB    P.AWCK(R5)      ;LOAD WRITE CHECK AFTER WRITE
 4933  020426  012765  000012  000100  P6$:    MOV     #D.CERT,P.CERT(R5) ;LOAD CORRECTABLE READ ERROR THRESHOLD
 4934  020434  012765  000005  000102  P7$:    MOV     #D.UERT,P.UERT(R5) ;LOAD UNCORRECTABLE READ ERROR THRESHOLD
 4935  020442  012765  000005  000104  P8$:    MOV     #D.SKET,P.SKET(R5) ;LOAD SEEK ERROR THRESHOLD
 4936  020450  105065  000120          P9$:    CLRB    P.IERC(R5)      ;CLEAR INHIBIT ERROR CORRECTION
 4937  020454  012765  077777  000110  P10$:   MOV     #D.CTRH,P.MXCD+2(R5) ;LOAD COMMAND THRESHOLD
 4938  020462  012765  177770  000106          MOV     #D.CTRL,P.MXCD(R5)
 4939  020470  012765  077777  000114  P11$:   MOV     #D.WTHI,P.MXWT+2(R5) ;LOAD WORD TRANSFERED THRESHOLD
 4940  020476  012765  177770  000112          MOV     #D.WTLO,P.MXWT(R5)
 4941  020504  105065  000116          P12$:   CLRB    P.SMPL(R5)      ;MAKE SAMPLED COMPARES
 4942
 4943  020510  012603                          MOV     (SP)+,R3        ;RESTORE R3
 4944  020512  012604                          MOV     (SP)+,R4        ;RESTORE R4
 4945  020514  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
                                ;*
                                ;;*****************************************************************

020516  104403                  TSTDEF: RDLIN                   ;GET RESPONSE
020520  012603                          MOV     (SP)+,R3        ;STORE BUFFER ADDRESS
020522  121327  000054                  CMPB    (R3),#',        ;CHECK IF COMMA
020526  001415                          BEQ     10$             ;YES, RETURN
020530  005720                          TST     (R0)+           ;ADJUST R0
020532  105713                          TSTB    (R3)            ;CHECK IF CARRIAGE RETURN
020534  001412                          BEQ     10$             ;YES, RETURN
020536  005720                          TST     (R0)+           ;ADJUST R0
020540  122713  000032                  CMPB    #32,(R3)        ;CHECK IF CONTROL Z <^Z>
020544  001406                          BEQ     10$             ;YES, RETURN
020546  005720                          TST     (R0)+           ;ADJUST R0
020550  122713  000003                  CMPB    #3,(R3)         ;CHECK IF CONTROL C <^C>
020554  001402                          BEQ     10$             ;YES, RETURN
020556  005720                          TST     (R0)+           ;ADJUST R0
020560  000200                          RTS     R0              ;RETURN

020562  011000                  10$:    MOV     (R0),R0         ;STORE RETURN ADDRESS
020564  000200                          RTS     R0              ;RETURN

020566  104401  061064          DTYPE:  TYPE    ,PAR022         ;TYPE "TESTING RK06? Y OR N"
020572  004037  020516                  JSR     R0,TSTDEF       ;GET INPUT
020576  020622                          1$                      ;COMMA
020600  020622                          1$                      ;CR
020602  020622                          1$                      ;CONT Z
020604  020622                          1$                      ;CONT C

020606  121327  000131                  CMPB    (R3),#'Y        ;SEE IF RK06
020612  001406                          BEQ     2$              ;BR IF YES
020614  121327  000116                  CMPB    (R3),#'N        ;SEE IF RK07
020620  001412                          BEQ     3$              ;BR IF YES

020622  104401  056440          1$:     TYPE    ,ILLCMD         ;ELSE ERROR
020626  000757                          BR      DTYPE           ;TRY AGAIN

020630  152765  000004  000007  2$:     BISB    #B.CDT,P.CS1H(R5)       ;RK06
020636  012737  000632  027332          MOV     #632,LSTCYL             ;LAST CYLINDER ADDRESS
020644  000207                          RTS     PC
020646  152765  000004  000007  3$:     BISB    #B.CDT,P.CS1H(R5)       ;RK07
020654  012737  001456  027332          MOV     #1456,LSTCYL            ;LAST CYL ADDRESS
```

```
 5002  020662  000207                        RTS     PC
 5003
 5004
```

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

020664  012037  020672          LDEXAR: MOV     (R0)+,2$        ;LOAD NUMBER OF EXCULSION AREA

020670  104401                  1$:     TYPE                    ;TYPE 'NTH EXCLUDED AREA =''
020672  000000                  2$:     .WORD   0
020674  104401  060650                  TYPE    ,PAR018
020700  004037  020516                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
020704  021310                          30$                     ;COMMA DETECTED
020706  021332                          45$                     ;CARRIAGE RETURN DETECTED
020710  021332                          45$                     ;CONTROL Z <^Z> DETECTED
020712  021330                          40$                     ;CONTROL C <^C> DETECTED
020714  010346                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
020716  004737  054406                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
020722  021310                          30$                     ;ERROR RETURN
020724  023716  027332                  CMP     LSTCYL,(SP)     ;CHECK IF LEGAL CYLINDER
020730  103566                          BLO     29$             ;NO, TRY AGAIN
020732  011637  001616                  MOV     (SP),EXAREA     ;MULTIPLY CYLINDER ADDRESS BY 3
020736  006316                          ASL     (SP)
020740  062637  001616                  ADD     (SP)+,EXAREA
020744  010304                          MOV     R3,R4           ;STORE BEGINNING OF COMMAND STRING

020746  105714                  64$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
020750  001525                          BEQ     10$             ;YES, EXIT
```

```
5061  020752  122427  000054            CMPB    (R4)+,#',       ;CHECK FOR COMMA
5062  020756  001373                    BNE     64$             ;NO, GO TEST NEXT CHARACTER
5063  020760  105714                    TSTB    (R4)            ;CHECK IF CARRIAGE RETURN
5064  020762  001552                    BEQ     30$             ;YES, TRY AGAIN
5065  020764  121427  000054            CMPB    (R4),#',        ;CHECK IF COMMA
5066  020770  001547                    BEQ     30$             ;YES, TRY AGAIN
5067  020772  010446                    MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
5068  020774  004737  054406            JSR     PC,OCTBIN       ;CONVERT TO BINARY
5069  021000  021310                    30$                     ;ERROR RETURN
5070  021002  022716  000002            CMP     #2,(SP)         ;CHECK IF LEGAL TRACK
5071  021006  103537                    BLO     29$             ;TRY AGAIN
5072  021010  063716  001616            ADD     EXAREA,(SP)     ;PUT MULTIPLIER ON STACK
5073  021014  012746  000026            MOV     #22.,-(SP)      ;PUT MULTIPLIER ON STACK
5074  021020  004737  055210            JSR     PC,$MULT        ;DO MULTIPLICATION
5075  021024  005216                    INC     (SP)            ;ADD 1
5076  021026  012637  001616            MOV     (SP)+,EXAREA    ;STORE VALUE
5077  021032  005726                    TST     (SP)+           ;THROW AWAY MOST SIGNIFICANT BITS
5078
5079  021034  105714            65$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
5080  021036  001511                    BEQ     20$             ;YES, EXIT
5081  021040  122427  000054            CMPB    (R4)+,#',       ;CHECK FOR COMMA
5082  021044  001373                    BNE     65$             ;NO, GO TEST NEXT CHARACTER
5083  021046  105714                    TSTB    (R4)            ;CHECK IF CARRIAGE RETURN
5084  021050  001517                    BEQ     30$             ;YES, TRY AGAIN
5085  021052  121427  000054            CMPB    (R4),#',        ;CHECK IF COMMA
5086  021056  001514                    BEQ     30$             ;YES, TRY AGAIN
5087  021060  010446                    MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
5088  021062  004737  054406            JSR     PC,OCTBIN       ;CONVERT TO BINARY
5089  021066  021310                    30$                     ;ERROR RETURN
5090  021070  023716  027332            CMP     LSTCYL,(SP)     ;CHECK IF LEGAL CYLINDER
5091  021074  103504                    BLO     29$             ;NO, TRY AGAIN
5092  021076  011637  001620            MOV     (SP),EXAREA+2   ;MULTIPLY CYLINDER ADDRESS BY 3
5093  021102  006316                    ASL     (SP)
5094  021104  062637  001620            ADD     (SP)+,EXAREA+2
5095
5096  021110  105714            66$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
5097  021112  001476                    BEQ     30$             ;YES, EXIT
5098  021114  122427  000054            CMPB    (R4)+,#',       ;CHECK FOR COMMA
5099  021120  001373                    BNE     66$             ;NO, GO TEST NEXT CHARACTER
5100  021122  105714                    TSTB    (R4)            ;CHECK IF CARRIAGE RETURN
5101  021124  001471                    BEQ     30$             ;YES, TRY AGAIN
5102  021126  121427  000054            CMPB    (R4),#',        ;CHECK IF COMMA
5103  021132  001466                    BEQ     30$             ;YES, TRY AGAIN
5104  021134  010446                    MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
5105  021136  004737  054406            JSR     PC,OCTBIN       ;CONVERT TO BINARY
5106  021142  021310                    30$                     ;ERROR RETURN
5107  021144  022716  000002            CMP     #2,(SP)         ;CHECK IF LEGAL TRACK
5108  021150  103456                    BLO     29$             ;TRY AGAIN
5109  021152  063716  001620            ADD     EXAREA+2,(SP)   ;PUT MULTIPLIER ON STACK
5110  021156  012746  000026            MOV     #22.,-(SP)      ;PUT MULTIPLIER ON STACK
5111  021162  004737  055210            JSR     PC,$MULT        ;DO MULTIPLICATION
5112  021166  005216                    INC     (SP)            ;ADD 1
5113  021170  012637  001620            MOV     (SP)+,EXAREA+2  ;STORE VALUE
5114  021174  005726                    TST     (SP)+           ;THROW AWAY MOST SIGNIFICANT BITS
5115  021176  023737  001616  001620    CMP     EXAREA,EXAREA+2 ;CHECK IF LEGAL RANGE
5116  021204  103041                    BHIS    30$             ;NO, TRY AGAIN
```

```
 5117  021206  013721  001616            MOV     EXAREA,(R1)+    ;LOAD EXCLUDED AREA
 5118  021212  013721  001620            MOV     EXAREA+2,(R1)+
 5119  021216  062700  000004            ADD     #4,R0           ;ADJUST R0
 5120  021222  000200                    RTS     R0              ;RETURN
 5121
 5122  021224  013746  001616     10$:   MOV     EXAREA,-(SP)    ;LOAD MULTIPLIER
 5123  021230  012746  000026            MOV     #22.,-(SP)      ;LOAD MULTICAND
 5124  021234  004737  055210            JSR     PC,$MULT        ;CALCULATE VALUE
 5125  021240  005216                    INC     (SP)            ;INCREMENT RESULTS
 5126  021242  011621                    MOV     (SP),(R1)+      ;STORE EXCLUDED AREA
 5127  021244  062716  000102            ADD     #66.,(SP)
 5128  021250  012621                    MOV     (SP)+,(R1)+
 5129  021252  005726                    TST     (SP)+           ;THROW AWAY MUST SIGNIFICANT BITS
 5130  021254  062700  000004            ADD     #4,R0           ;ADJUST R0
 5131  021260  000200                    RTS     R0              ;RETURN
 5132
  133  021262  013721  001616     20$:   MOV     EXAREA,(R1)+    ;STORE EXCLUDEED AREA
 5134  021266  062737  000026  001616    ADD     #22.,EXAREA
 5135  021274  013721  001616            MOV     EXAREA,(R1)+
 5136  021300  062700  000004            ADD     #4,R0           ;ADJUST R0
 5137  021304  000200                    RTS     R0              ;RETURN
 5138
 5139  021306  005726             29$:   TST     (SP)+           ;ADJUST STACK
 5140  021310  010337  021316     30$:   MOV     R3,31$          ;LOAD BUFFER ADDRESS
 5141  021314  104401                    TYPE                    ;TYPE RECEIVED INPUT
 5142  021316  000000             31$:   .WORD   0               ;BUFFER ADDRESS
 5143  021320  104401  001164            TYPE    ,$QUES          ;TYPE QUESTION MARK
 5144  021324  000137  020670            JMP     1$              ;TRY AGAIN
 5145
 5146  021330  005720             40$:   TST     (R0)+           ;ADJUST R0
 5147  021332  011000             45$:   MOV     (R0),R0         ;STORE SPECIAL RETURN ADDRESS
 5148  021334  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
                        ;*
                        ;;*****************************************************************

021336  005065  000146          GENERT: CLR     P.RECT(R5)      ;CLEAR ERROR FLAGS AND
021342  005065  000150                  CLR     P.DSTT(R5)      ;  RETRY COUNTS
021346  005065  000212                  CLR     P.ERR(R5)

021352  032777  040000  157560          BIT     #SW14,@SWR      ;CHECK IF NEW COMMAND
                                                                ;  SHOULD BE GENERATED
021360  001454                          BEQ     10$             ;YES, GENERATE NEW COMMAND
021362  105765  000211                  TSTB    P.SEEK(R5)      ;CHECK IF PREVIOUS CYLINDER ISSUED
021366  001020                          BNE     1$              ;YES, SET UP DATA TRANSFER
021370  112765  177777  000211          MOVB    #-1,P.SEEK(R5)  ;INDICATE SEEK HAS BEEN ISSUED
021376  112765  000117  000001          MOVB    #SEEK,P.CMND(R5)  ;LOAD SEEK PARAMETERS
021404  016565  000136  000002          MOV     P.LCYL(R5),P.CYLN(R5)  ;LOAD PRIOR CYLINDER
021412  016565  000140  000004          MOV     P.LSEC(R5),P.SECT(R5)  ;LOAD PRIOR SECTOR AND TRACK
021420  004037  051564                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON
021424  001324                          CINITQ                  ;  COMMAND INITIATION QUEUE
021426  000207                          RTS     PC              ;RETURN
```

```
5205  021430  105065  000211          1$:     CLRB    P.SEEK(R5)          ;INDICATE DATA TRANSFER ISSUED
5206  021434  116565  000122  000121          MOVB    P.RDPT(R5),P.DPAT(R5)  ;LOAD LAST DATA PATTERN
5207  021442  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5)  ;LOAD LAST COMMAND
5208  021450  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5)  ;LOAD LAST CYLINDER ADDRESS
5209  021456  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)  ;CHECK IF WRITE CHECK
5210  021464  001003                          BNE     2$                  ;NO, DO NOT SET FLAG
5211  021466  052765  000200  000014          BIS     #W.WCK,P.PRST(R5)   ;SET ISSUE WRITE FOR WRITE CHECK
5212  021474  016565  000126  000004  2$:     MOV     P.RSEC(R5),P.SECT(R5)  ;LOAD LAST SECTOR AND TRACK
5213  021502  016565  000130  000012          MOV     P.RWC(R5),P.WC(R5)  ;LOAD LAST WORD COUNT
5214  021510  000207                          RTS     PC                  ;RETURN
5215
5216  021512  005077  157426          10$:    CLR     @$TKS               ;LOCK OUT TTY INTERRUPTS
5217  021516  010446                          MOV     R4,-(SP)            ;STORE R4 ON STACK
5218  021520  010346                          MOV     R3,-(SP)            ;STORE R3 ON STACK
5219  021522  010146                          MOV     R1,-(SP)            ;STORE R1 ON STACK
5220  021524  010046                          MOV     R0,-(SP)            ;STORE R0 ON STACK
5221  021526  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5)  ;STORE LAST COMMAND
5222  021534  016565  000124  000136          MOV     P.RCYL(R5),P.LCYL(R5)  ;STORE LAST CYLINDER
5223  021542  016565  000126  000140          MOV     P.RSEC(R5),P.LSEC(R5)  ;STORE LAST SECTOR AND TRACK
5224  021550  016565  000130  000142          MOV     P.RWC(R5),P.LWC(R5)  ;STORE LAST WORD COUNT
5225  021556  116565  000122  000144          MOVB    P.RDPT(R5),P.LDPT(R5) ;STORE LAST PATTERN
5226  021564  004737  055324                  JSR     PC,$RAND            ;GENERATE RANDOM NUMBER
5227  021570  013746  055424                  MOV     $LONUM,-(SP)        ;LOAD MULTIPLIER
5228  021574  013746  001354                  MOV     MAXBUF,-(SP)        ;LOAD MULTIPLICAND
5229  021600  005216                          INC     (SP)                ;INCREMENT WORD COUNT MAX.
5230  021602  004737  055210                  JSR     PC,$MULT            ;CALCULATE WORD COUNT
5231  021606  005726                          TST     (SP)+               ;THROW AWAY LEAST SIGNIFICANT BITS
5232  021610  042716  100003                  BIC     #100003,(SP)        ;MAKE DIVISIBLE BY 4 AND < 32K
5233  021614  001002                          BNE     11$                 ;IF NOT ZERO USE WORD COUNT
5234  021616  012716  000004                  MOV     #4,(SP)             ;MAKE WORD COUNT 4
5235  021622  116604  000001          11$:    MOVB    1(SP),R4            ;CALCULATE NUMBER OF SECTORS TO
5236  021626  105716                          TSTB    (SP)                ;  TRANSFERRED
5237  021630  001401                          BEQ     12$                 ;IF EVEN SECTORS DO NOT MODIFY
5238  021632  005204                          INC     R4                  ;  SECTOR COUNT
5239  021634  011646          12$:    MOV     (SP),-(SP)          ;SAVE WORD COUNT
5240  021636  005416                          NEG     (SP)                ;GENERATE NEGATIVE NUMBER
5241  021640  012665  000012                  MOV     (SP)+,F.WC(R5)      ;LOAD PRESENT WORD COUNT
5242  021644  062765  000001  000152          ADD     #1,P.NODF(R5)       ;INCREMENT NUMBER OF ORDERS
5243  021652  005565  000154                  ADC     P.NODR+2(R5)
5244  021656  113746  055422                  MOVB    $HINUM,-(SP)        ;RANDOM NUMBER FOR COMMAND
5245  021662  042716  177770                  BIC     #177770,(SP)        ;KEEP LOW 3 BITS
5246  021666  126526  000076                  CMPB    P.RATE(R5),(SP)+    ;DETERMINE IF READ OR WRITE
5247  021672  101462                          BLOS    15$                 ;ISSUE READ
5248  021674  113746  055423                  MOVB    $HINUM+1,-(SP)      ;RANDOM NUMBER FOR PATTERN
5249  021700  042716  177760                  BIC     #177760,(SP)        ;KEEP LOW 4 BITS
5250  021704  111665  000122                  MOVB    (SP),P.RDPT(R5)     ;STORE DATA PATTERN
5251  021710  112665  000121                  MOVB    (SP)+,P.DPAT(R5)
5252  021714  105765  000077                  TSTB    P.AWCK(R5)          ;CHECK IF AUTO WRITE CHECK
5253  021720  001017                          BNE     13$                 ;YES, ISSUE WRITE CHECK
5254  021722  113746  055422                  MOVB    $HINUM,-(SP)        ;RANDOM NUMBER FOR WRITE CHECK
5255  021726  042726  177707                  BIC     #177707,(SP)+       ;CHECK IF WRITE COMMAND
5256  021732  001412                          BEQ     13$                 ;YES, ISSUE WRITE CHECK
5257  021734  112765  000123  000123          MOVB    #WRDATA,P.RCMD(R5)  ;ISSUE WRITE DATA
5258  021742  062665  000156                  ADD     (SP)+,P.NWRT(R5)    ;ADD NUMBER OF WORDS TO BE WRITTEN
5259  021746  005565  000160                  ADC     P.NWRT+2(R5)
5260  021752  005565  000162                  ADC     P.NWRT+4(R5)
```

```
 5261  021756  000441                          BR      18$
 5262
 5263  021760  112765  000131  000123  13$:    MOVB    #WRTCHK,P.RCMD(R5)  ;ISSUE WRITE CHECK
 5264  021766  052765  000200  000014          BIS     #W.WCK,P.PRST(R5)   ;SET NO WRITE FOR WRITE CHECK
 5265  021774  062765  000001  000152          ADD     #1,P.NODR(R5)    ;INCREMENT NUMBER OF COMMAND ISSUED
 5266  022002  005565  000154                  ADC     P.NODR+2(R5)
 5267  022006  061665  000156                  ADD     (SP),P.NWRT(R5) ;ADD NUMBER OF WORDS TO BE WRITTEN
 5268  022012  005565  000160                  ADC     P.NWRT+2(R5)
 5269  022016  005565  000162                  ADC     P.NWRT+4(R5)
 5270  022022  062665  000164                  ADD     (SP)+,P.NRD(R5) ;ADD NUMBER OF WORDS READ
 5271  022026  005565  000166                  ADC     P.NRD+2(R5)
 5272  022032  005565  000170                  ADC     P.NRD+4(R5)
 5273  022036  000411                          BR      18$
 5274
 5275  022040  112765  000121  000123  15$:    MOVB    #RDDATA,P.RCMD(R5)  ;ISSUE READ DATA
 5276  022046  062665  000164                  ADD     (SP)+,P.NRD(R5) ;ADD NUMBER OF WORDS READ
 5277  022052  005565  000166                  ADC     P.NRD+2(R5)
 5278  022056  005565  000170                  ADC     P.NRD+4(R5)
 5279  022062  116565  000123  000001  18$:    MOVB    P.RCMD(R5),P.CMND(R5)  ;LOAD COMMAND
 5280  022070  012700  000100                  MOV     #100,R0          ;LOAD REGENERATE COUNT
 5281
 5282  022074  010503                  20$:    MOV     R5,R3            ;LOAD INDEX
 5283  022076  062703  000232                  ADD     #P.EXAR,R3       ;CALCULATE EXCLUDED AREA INDEX
 5284  022102  012701  000005                  MOV     #5,R1            ;LOAD BAD TRACK COUNT
 5285  022106  004737  055324                  JSR     PC,$RAND         ;GENERATE RANDOM NUMBER
 5286  022112  016546  000070                  MOV     P.MXCL(R5),-(SP)  ;STORE MAXIMUM CYLINDER
 5287  022116  166516  000066                  SUB     P.MNCL(R5),(SP) ;CALCULATE DIFFERENCE
 5288  022122  001406                          BEQ     21$              ;IF 0, SKIP MULTIPLY
 5289  022124  005216                          INC     (SP)             ;INCREMENT DIFFERENCE
 5290  022126  013746  055424                  MOV     $LONUM,-(SP)     ;PUT MULTICAND ON STACK
 5291  022132  004737  055210                  JSR     PC,$MULT         ;CALCULATE CYLINDER ADDRESS
 5292  022136  005726                          TST     (SP)+            ;THROW AWAY LEASE SIGNIFICANT BITS
 5293
 5294  022140  066516  000066          21$:    ADD     P.MNCL(R5),(SP) ;CALCULATE CYLINDER
 5295  022144  011665  000124                  MOV     (SP),P.RCYL(R5) ;STORE CYLINDER
 5296  022150  116537  000073  001572          MOVB    P.MXTR(R5),TMPTRK  ;STORE MAXIMUM TRACK
 5297  022156  116537  000072  001574          MOVB    P.MNTR(R5),TMPTRK+2  ;STORE MINIMUM TRACK
 5298  022164  163737  001574  001572          SUB     TMPTRK+2,TMPTRK ;CALCULATE PERFERENCE
 5299  022172  001415                          BEQ     22$              ;IF=0, SKIP MULTIPLY
 5300  022174  005046                          CLR     -(SP)            ;MAKE ROOM ON STACK
 5301  022176  113716  055423                  MOVB    $HINUM+1,(SP)    ;PUT IN MULTIPLIER ON STACK
 5302  022202  013746  001572                  MOV     TMPTRK,-(SP)     ;PUT TRACK NUMBER ON STACK
 5303  022206  005216                          INC     (SP)             ;INCREMENT DIFFERENCE
 5304  022210  004737  055210                  JSR     PC,$MULT         ;CALCULATE TRACK
 5305  022214  116637  000001  001572          MOVB    1(SP),TMPTRK     ;STORE TRACK
 5306  022222  062706  000004                  ADD     #4,SP            ;ADJUST STACK
 5307
 5308  022226  063737  001574  001572  22$:    ADD     TMPTRK+2,TMPTRK ;CALCULATE TRACK
 5309  022234  113765  001572  000127          MOVB    TMPTRK,P.RTRK(R5)  ;STORE TRACK NUMBER
 5310  022242  116537  000075  001576          MOVB    P.MXSC(R5),TMPSEC  ;STORE MAXIMUM SECTOR
 5311  022250  116537  000074  001600          MOVB    P.MNSC(R5),TMPSEC+2  ;STORE MINIMUM SECTOR
 5312  022256  163737  001600  001576          SUB     TMPSEC+2,TMPSEC ;CALCULATE DIFFERENCE
 5313  022264  001415                          BEQ     23$              ;IF=0, SKIP MULTIPLY
 5314  022266  005046                          CLR     -(SP)            ;MAKE ROOM ON STACK
 5315  022270  113716  055422                  MOVB    $HINUM,(SP)      ;PUT IN MULTIPLIER
 5316  022274  013746  001576                  MOV     TMPSEC,-(SP)     ;PUT SECTOR NUMBER ON STACK
```

```
5317  022300  005216                        INC     (SP)              ;INCREMENT DIFFERENCE
5318  022302  004737  055210                JSR     PC,$MULT          ;CALCULATE SECTOR
5319  022306  116637  000001  001576        MOVB    1(SP),TMPSEC      ;STORE SECTOR
5320  022314  062706  000004                ADD     #4,SP             ;ADJUST STACK
5321
5322  022320  063737  001600  001576 23$:   ADD     TMPSEC+2,TMPSEC ;CALCULATE SECTOR
5323  022326  113765  001576  000126        MOVB    TMPSEC,P.RSEC(R5)  ;STORE SECTOR NUMBER
5324  022332  006316                        ASL     (SP)              ;MULTIPLY CYLINDER BY 3
5325  022336  066516  000124                ADD     P.RCYL(R5),(SP)
5326  022342  063716  001572                ADD     TMPTRK,(SP)       ;ADD TRACK
5327  022346  012746  000026                MOV     #22.,-(SP)        ;MULTIPLICAND - 22 SECTORS/TRACK
5328  022352  004737  055210                JSR     PC,$MULT
5329  022356  012616                        MOV     (SP)+,(SP)        ;THROW AWAY MOST SIGNIFICANT BITS
5330  022360  063716  001576                ADD     TMPSEC,(SP)       ;ADD SECTOR
5331  022364  005216                        INC     (SP)              ;CONSTANT FOR BAD TRACKS
5332  022366  061604                        ADD     (SP),R4           ;ADD NUMBER OF SECTORS
5333  022370  004737  027252                JSR     PC,GETNUM
5334  022374  023704  027340                CMP     HOLD1,R4          ;CHK IF DATA XFER EXEEDS DISK
5335  022400  103421                        BLO     40$               ;CALCULATE NEW DISK ADDRESS
5336
5337  022402  005713                25$:    TST     (R3)              ;CHECK IF BAD TRACK INTERVAL
5338                                                                  ;  EXIST
5339  022404  001407                        BEQ     30$               ;NO, ISSUE COMMAND
5340  022406  021623                        CMP     (SP),(R3)+        ;CHECK IF IN BAD TRACK INTERVAL
5341  022410  103402                        BLO     26$               ;NO, CHECK NEXT BAD TRACK 1
5342  022412  020413                        CMP     R4,(R3)
5343  022414  103413                        BLO     40$               ;YES, CALCULATE NEW DISK ADDRESS
5344  022416  005723                26$:    TST     (R3)+             ;ADJUST R3
5345  022420  005301                        DEC     R1                ;DECREMENT BAD BLOCK COUNT
5346  022422  001367                        BNE     25$               ;IF NOT 0 GET NEXT BAD BLOCK
5347
5348  022424  005726                30$:    TST     (SP)+             ;ADJUST STACK
5349  022426  016565  000124  000002        MOV     P.RCYL(R5),P.CYLN(R5)  ;LOAD CYLINDER
5350  022434  016565  000126  000004        MOV     P.RSEC(R5),P.SECT(R5)  ;LOAD TRACK AND SECTOR
5351  022442  000416                        BR      45$               ;RETURN
5352
5353  022444  162604                40$:    SUB     (SP)+,R4          ;GET NUMBER OF SECTOR TRANSFERRED
5354  022446  005300                        DEC     R0                ;DECREMENT REGENERATE COUNT
5355  022450  001402                        BEQ     41$               ;DROP DRIVE CANNOT GENERATE
5356                                                                  ;  RANDOM DISK ADDRESS
5357  022452  000137  022074                JMP     20$               ;GET ANOTHER DISK ADDRESS
5358
5359  022456  013737  001452  177776 41$:   MOV     RKPRI,PS          ;LOCK OUT RK06 AND TTY INERRUPTS
5360  022464  104401  060705                TYPE    ,PAR020           ;TYPE 'UNABLE TO GENERATE NEW COMMAND''
5361  022470  004737  013162                JSR     PC,DROP           ;DROP DRIVE
5362  022474  005037  177776                CLR     PS                ;ALLOW RK06 INTERRUPTS
5363
5364  022500  012600                45$:    MOV     (SP)+,R0          ;RESTORE R0
5365  022502  012601                        MOV     (SP)+,R1          ;RESTORE R1
5366  022504  012603                        MOV     (SP)+,R3          ;RESTORE R3
5367  022506  012604                        MOV     (SP)+,R4          ;RESTORE R4
5368  022510  012777  000100  156426        MOV     #IE,@$TKS         ;ALLOW TTY INTERRUPTS
5369  022516  000207                        RTS     PC                ;RETURN
```

```
5370                                    .SBTTL  GET AVAILIBLE BUFFER ROUTINE
5371
5372  022520  013746  177776    GETBUF: MOV     PS,-(SP)        ;STORE PSW ON STACK
5373  022524  013737  001452  177776    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
5374  022532  013705  001320            MOV     BWAITQ,R5       ;CHECK IF COMMAND WAITING FOR BUFFER
5375  022536  001424                    BEQ     10$             ;RETURN IF QUEUE IS EMPTY
5376  022540  004037  022616    1$:     JSR     R0,BUFFAL       ;ALLOCATE BUFFER
5377  022544  022610                    10$                     ;NOT ENOUGH MEMORY RETURN
5378  022546  004037  051644            JSR     R0,Q.POP        ;GET NEXT PARMETER BLOCK FROM
5379  022552  001320                    BWAITQ                  ;  BUFFER WAIT QUEUE
5380  022554  005726                    TST     (SP)+           ;ADJUST STACK
5381  022556  005037  177776            CLR     PS              ;ALLOW RK06 INTERRUPTS
5382  022562  004737  023234            JSR     PC,LDDATA       ;LOAD BUFFER
5383  022566  004037  051564            JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON
5384  022572  001324                    CINITQ                  ;  COMMAND INITIATION QUEUE
5385  022574  013737  001452  177776    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
5386  022602  013705  001320            MOV     BWAITQ,R5       ;CHECK IF COMMAND WAITING FOR BUFFER
5387  022606  001354                    BNE     1$              ;YES, TRY TO ALLOCATE BUFFER
5388
5389  022610  012637  177776    10$:    MOV     (SP)+,PS        ;RESTORE PSW
5390  022614  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>
                                ;*
                                ;;******************************************************************

022616  010446          BUFFAL: MOV     R4,-(SP)        ;STORE R4 ON STACK
022620  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
022622  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
022624  016501  000012          MOV     P.WC(R5),R1     ;STORE WORD COUNT
022630  010165  000130          MOV     R1,P.RWC(R5)    ;STORE WORD COUNT
022634  005401                  NEG     R1              ;GET ACTUAL WORD COUNT
022636  006301                  ASL     R1              ;MULTIPLY WORD COUNT BY 2
022640  013704  001670          MOV     FBLKC,R4        ;STORE FREE BLOCK TABLE ADDRESS
022644  001420                  BEQ     2$              ;RETURN, IF NO BLOCKS AVAILABLE
022646  012703  001672          MOV     #FBLKT,R3       ;LOAD ADDRESS OF FREE BLOCK TABLE
022652  022704  000011          CMP     #9.,R4          ;HANG IF FREE BLOCK COUNT IS GREATER
022656  101001                  BHI     1$              ;  OR EQUAL TO 9
022660  000000                  HALT

022662  012346          1$:     MOV     (R3)+,-(SP)     ;STORE BEGINNING OF FREE BLOCK
022664  060116                  ADD     R1,(SP)         ;ADD WORD COUNT
022666  103411                  BCS     5$              ;CHECK IF ADDRESS OVERFLOW
022670  021316                  CMP     (R3),(SP)       ;CHECK IF ENOUGH ROOM
022672  001423                  BEQ     15$             ;YES, ELIMINATE BLOCK
022674  101010                  BHI     10$             ;YES, SHORTEN BLOCK
022676  005723                  TST     (R3)+           ;ADJUST R3
022700  005726                  TST     (SP)+           ;ADJUST STACK
022702  005304                  DEC     R4              ;DECREMENT FREE BLOCK COUNT
022704  001366                  BNE     1$              ;EXAMINE NEXT FREE BLOCK
022706  011000          2$:     MOV     (R0),R0         ;LOAD RETURN ADDRESS
022710  000437                  BR      25$             ;RETURN

022712  005726          5$:     TST     (SP)+           ;ADJUST STACK
022714  000774                  BR      2$              ;RETURN

022716  014365  000132  10$:     MOV     -(R3),P.RBAL(R5) ;LOAD BUS ADDRESS
022722  011365  000010          MOV     (R3),P.BALO(R5) ;LOAD BUS ADDRESS
022726  012613                  MOV     (SP)+,(R3)      ;LOAD NEW BASE
022730  005720                  TST     (R0)+           ;ADJUST RETURN
022732  112765  177777  000145  MOVB    #-1,P.BUFF(R5)  ;INDICATE THAT BUFFER HAS BEEN ASSIGNED
```

```
5447  022740  000423                        BR      25$             ;RETURN
5448
5449  022742  010301                15$:    MOV     R3,R1           ;STORE WORD COUNT
5450  022744  014365  000132                MOV     -(R3),P.RBAL(R5)
5451  022750  011365  000010                MOV     (R3),P.BALO(R5) ;LOAD BUS ADDRESS
5452  022754  005726                        TST     (SP)+           ;ADJUST STACK
5453  022756  005721                        TST     (R1)+           ;ADJUST R1
5454  022760  005720                        TST     (R0)+           ;ADJUST RETURN
5455  022764  112765  177777  000145        MOVB    #-1,P.BUFF(R5)  ;INDICATE THAT BUFFER HAS BEEN ASSIGNED
5456  022770  005337  001670                DEC     FBLKC           ;DECREMENT FREE BLOCK COUNT
5457  022774  005304                        DEC     R4              ;DECREMENT REMAIN BLOCK COUNT
5458  022776  001404                        BEQ     25$             ;IF 0, RETURN
5459
5460  023000  012123                20$:    MOV     (R1)+,(R3)+     ;TRANSFER TWO WORDS
5461  023002  012123                        MOV     (R1)+,(R3)+
5462  023004  005304                        DEC     R4              ;DECREMENT REMAINDING BLOCKS TO BE ADJUSTED
5463  023006  001374                        BNE     20$             ;IF NOT 0, TRANSFER NEXT TWO WORDS
5464  023010  012601                25$:    MOV     (SP)+,R1        ;RESTORE R1
5465  023012  012603                        MOV     (SP)+,R3        ;RESTORE R3
5466  023014  012604                        MOV     (SP)+,R4        ;RESTORE R4
5467  023016  000200                        RTS     R0              ;RETURN
5468
```

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

023020  010446          BUFREL: MOV     R4,-(SP)        ;STORE R4 ON STACK
023022  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
023024  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
023026  010046                  MOV     R0,-(SP)        ;STORE R0 ON STACK
023030  105765  000145          TSTB    P.BUFF(R5)      ;CHECK IF BUFFER ASSIGNED
023034  001472                  BEQ     25$             ;NO, RETURN
023036  105065  000145          CLRB    P.BUFF(R5)      ;CLEAR BUFFER ALLOCATED
023042  016501  000130          MOV     P.RWC(R5),R1    ;STORE ALLOCATED WORD COUNT
023046  005401                  NEG     R1              ;GET ACTUAL WORD COUNT
023050  006301                  ASL     R1              ;MULTIPLY BY 2
023052  016546  000132          MOV     P.RBAL(R5),-(SP)  ;STORE BUFFER ADDRESS
023056  061601                  ADD     (SP),R1         ;CALCULATE FINAL ADDRESS
023060  012703  001672          MOV     #FBLKT,R3       ;LOAD FREE BLOCK TABLE ADDRESS
023064  013704  001670          MOV     FBLKC,R4        ;STORE FREE BLOCK COUNT
023070  001424                  BEQ     8$              ;ADD FREE BLOCK

023072  022301          1$:     CMP     (R3)+,R1        ;COMPARE END OF BUFFER AND FREE BLOCK
023074  001427                  BEQ     10$             ;EQUAL MERGE BLOCKS
023076  101005                  BHI     5$              ;LESS THEN ADD FREE BLOCK
023100  022316                  CMP     (R3)+,(SP)      ;COMPARE BEGINNING OF BUFFER
                                                        ;  AND END OF FREE BLOCK
023102  001426                  BEQ     15$             ;EQUAL MERGE
023104  005304                  DEC     R4              ;DECREMENT FREE BLOCK COUNT
023106  001371                  BNE     1$              ;GO TO NEXT FREE BLOCK
023110  000414                  BR      8$              ;GO ADD BLOCK

023112  010446          5$:     MOV     R4,-(SP)        ;PUT REMAINING COUNT ON STACK
023114  006316                  ASL     (SP)            ;MULTIPLY BY 4
023116  006316                  ASL     (SP)
023120  005743                  TST     -(R3)           ;DECREMENT R3 BY 2
023122  062603                  ADD     (SP)+,R3        ;CALCULATE OLD END OF FREE BLOCK TABLE
023124  010300                  MOV     R3,R0
023126  062700  000004          ADD     #4,R0           ;CALCULATE NEW END OF FREE BLOCK TABLE
023132  014340          6$:     MOV     -(R3),-(R0)     ;MOVE TABLE
```

```
 5525  023134  014340                    MOV     -(R3),-(R0)
 5526  023136  005304                    DEC     R4              ;DECREMENT FREE BLOCK COUNT
 5527  023140  001374                    BNE     6$              ;IF NOT ZERO, COUNT
 5528
 5529  023142  005237  001670      8$:   INC     FBLKC           ;INCREMENT FREE BLOCK COUNT
 5530  023146  012623                    MOV     (SP)+,(R3)+     ;STORE BEGINNING OF FREE BLOCK
 5531  023150  010123                    MOV     R1,(R3)+        ;STORE END OF FREE BLOCK
 5532  023152  000423                    BR      25$             ;RETURN
 5533
 5534  023154  012643            10$:    MOV     (SP)+,-(R3)     ;LOAD BEGINNING OF FREE BLOCK
 5535  023156  000421                    BR      25$             ;RETURN
 5536
 5537  023160  0057--            15$:    TST     (SP)+           ;ADJUST STACK
 5538  023162  005304                    DEC     R4              ;DECREMENT FREE BLOCKS LEFT
 5539  023164  001---                    BEQ     16$             ;NO MORE FREE BLOCK ADJUST END
 5540  023166  02130-                    CMP     (R3),R1         ;CHECK IF RELEASED BLOCK END=
```

```
5541
5542  023170  001402                        BEQ     20$             ;  FREE BLOCK BEGINNING
5543  023172  010143                16$:    MOV     R1,-(R3)        ;YES, MERGE BLOCKS
5544  023174  000412                        BR      25$             ;LOAD END OF FREE BLOCK
5545                                                                ;RETURN
5546  023176  010300                20$:    MOV     R3,R0           ;STORE R3 FOR BUFFER MOVEMENT
5547  023200  005743                        TST     -(R3)           ;ADJUST R3
5548  023202  005720                        TST     (R0)+           ;ADJUST R0
5549  023204  006304                        ASL     R4              ;MULTIPLY COUNT BY TWO
5550  023206  005304                        DEC     R4              ;DECREMENT TRANSFER COUNT
5551  023210  012023                21$:    MOV     (R0)+,(R3)+     ;MOVE TABLE ENTRY
5552  023212  005304                        DEC     R4              ;DECREMENT COUNT
5553  023214  001375                        BNE     21$             ;IF NOT 0, CONTINUE MOVE
5554  023216  005337  001670              . DEC     FBLKC           ;DECREMENT FREE BLOCK COUNT
5555
5556  023222  012600                25$:    MOV     (SP)+,R0        ;RESTORE R0
5557  023224  012601                        MOV     (SP)+,R1        ;RESTORE R1
5558  023226  012603                        MOV     (SP)+,R3        ;RESTORE R3
5559  023230  012604                        MOV     (SP)+,R4        ;RESTORE R4
5560  023232  000207                        RTS     PC              ;RETURN
5561
```

```
                        .SBTTL  LOAD DATA BUFFER

                        ;;*****************************************************************
                        ;*
                        ;*      THE REGISTERS BE USED AS FOLLOWS:
                        ;*
                        ;*      REGISTER                USE
                        ;*      --------                ---
                        ;*
                        ;*         R5           ADDRESS OF PARAMETER BLOCK
                        ;*         R4           BUFFER ADDRESS
                        ;*         R3           PATTERN ADDRESS
                        ;*         R2           WORDS LEFT IN SECTOR
                        ;*         R1           WORD COUNT
                        ;*         R0           PATTERN COUNT
                        ;*
                        ;*CALL  JSR     PC,LDDATA
                        ;*      RETURN
                        ;*
                        ;*
                        ;*      THE DATA WILL LOOK AS FOLLOWS:
                        ;*
                        ;*      *-----------------------*-----------*-----------*
                        ;*      !  CPU ID               !  PAT      !  PAT      ! WORD 0
                        ;*      ! NO. OF PAT. WORDS (N) ! NO. OF PAT. WORDS (N) ! WORD 1
                        ;*      !                RANDOM DATA                    ! WORDS 2
                        ;*      !                                               !  TO N+1
                        ;*      *-----------------------------------------------*
                        ;*      !                  ZERO FILL                    ! WORDS N+2
                        ;*      !                                               !  TO 255
                        ;*      *-----------------------------------------------*
                        ;*
                        ;;*****************************************************************

023234  010446          LDDATA: MOV     R4,-(SP)        ;STORE R4 ON STACK
023236  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
023240  010246                  MOV     R2,-(SP)        ;STORE R2 ON STACK
023242  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
023244  010046                  MOV     R0,-(SP)        ;STORE R0 ON STACK
023246  016504  000132          MOV     P.RBAL(R5),R4   ;GET BUS ADDRESS
023252  016501  000130          MOV     P.RWC(R5),R1    ;STORE WORD COUNT
023256  005401                  NEG     R1              ;GET POSITIVE NUMBER FOR WORD COUNT
023260  122765  000121  000001        CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ DATA
023266  001005                  BNE     5$              ;NO, LOAD BUFFER
023270  005024          3$:     CLR     (R4)+           ;CLEAR BUFFER BEFORE READ
023272  005301                  DEC     R1              ;CHECK IF DONE
023274  001375                  BNE     3$              ;NO, CONTINUE
023276  000137  023446          JMP     25$             ;RESTORE REGISTERS

023302  116500  000121  5$:     MOVB    P.DPAT(R5),R0   ;STORE PATTERN COUNT
023306  010046                  MOV     R0,-(SP)        ;STORE PATTERN FOR FIRST WORD
023310  006316                  ASL     (SP)            ;MULTIPLY PATTERN BY 4 FOR
023312  006316                  ASL     (SP)            ;  REDUNACY
023314  006316                  ASL     (SP)
023316  006316                  ASL     (SP)
023320  050016                  BIS     R0,(SP)         ;STORE FIRST WORD OF SECTOR
```

```
5618  023322  113766  001364  000001          MOVB    CPUID,1(SP)     ;  ON STACK
5619  023330  006300                          ASL     R0              ;MULTIPLY DATA PATTERN BY 2
5620  023332  016003  004602                  MOV     PATTAB(R0),R3   ;LOAD DATA PATTERN ADDRESS
5621  023336  012700  000040                  MOV     #32.,R0         ;LOAD PATTERN COUNT
5622  023342  011624                  11$:    MOV     (SP),(R4)+      ;STORE PATTERN AND CPU ID IN 1ST WORD
5623  023344  162701  000002                  SUB     #2,R1           ;DECREMENT WORD COUNT BY 2
5624  023350  022701  000376                  CMP     #254.,R1        ;CHECK IF REST OF SECTOR WILL BE
5625                                                                  ;  WRITTEN
5626  023354  101003                          BHI     12$             ;NO, USE R1 AS WORD COUNT
5627  023356  012746  000376                  MOV     #254.,-(SP)     ;STORE SECTOR COUNT = 254
5628  023362  000401                          BR      13$
5629
5630  023364  010146                  12$:    MOV     R1,-(SP)        ;STORE REMAINING WORD COUNT
5631  023366  111666  000001          13$:    MOVB    (SP),1(SP)      ;ADD REDUNDACY
5632  023372  012624                          MOV     (SP)+,(R4)+     ;STORE WORD COUNT IN 2ND WORD
5633  023374  012702  000376                  MOV     #254.,R2        ;LOAD SECTOR WORD COUNT
5634
5635  023400  012324                  15$:    MOV     (R3)+,(R4)+     ;STORE DATA PATTERN
5636  023402  005301                          DEC     R1              ;DECREMENT WORD COUNT
5637  023404  001417                          BEQ     20$             ;BRANCH IF FINISHED
5638  023406  005302                          DEC     R2              ;DECREMENT SECTOR COUNT
5639  023410  005300                          DEC     R0              ;DECREMENT PATTERN COUNT
5640  023412  001407                          BEQ     17$             ;IF ZERO, INITIALIZE PATTERN
5641  023414  005702                          TST     R2              ;ARE WE AT END OF SECTOR
5642  023416  001370                          BNE     15$             ;NO, CONTINUE TO LOAD DATA PATTERN
5643  023420  162703  000074                  SUB     #60.,R3         ;INITIALIZE PATTERN ADDRESS
5644  023424  012700  000040                  MOV     #32.,R0         ;INITIALIZE PATTERN COUNT
5645  023430  000744                          BR      11$             ;START NEXT SECTOR
5646
5647  023432  162703  000100          17$:    SUB     #64.,R3         ;INITIALIZE PATTERN ADDRESS
5648  023436  012700  000040                  MOV     #32.,R0         ;INITIALIZE PATTERN
5649  023442  000756                          BR      15$             ;REPEAT PATTERN
5650
5651  023444  005726                  20$:    TST     (SP)+           ;ADJUST STACK
5652  023446  012600                  25$:    MOV     (SP)+,R0        ;RESTORE R0
5653  023450  012601                          MOV     (SP)+,R1        ;RESTORE R1
5654  023452  012602                          MOV     (SP)+,R2        ;RESTORE R2
5655  023454  012603                          MOV     (SP)+,R3        ;RESTORE R3
5656  023456  012604                          MOV     (SP)+,R4        ;RESTORE R4
5657  023460  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  FIND FIRST DATA ERROR

023462  032777  000002  155450  BUFER1: BIT     #SW1,@SWR       ;CHECK IF INHIBIT SOFTWARE DATA
                                        ;  COMPARE
023470  001017                  BNE     5$              ;YES, RETURN
023472  013765  001362  000220  MOV     SOFCMP,P.CMLG(R5) ;LOAD COMPARISON LENGTH
023500  001413                  BEQ     5$              ;IF NO MORE COMPARISONS, RETURN
023502  016565  000010  000214  MOV     P.BALO(R5),P.ERHD(R5) ;LOAD ADDRESS OF START OF SECTOR
023510  016565  000012  000222  MOV     P.WC(R5),P.BFCP(R5) ;LOAD TRANSFER LENGTH
023516  005465  000222          NEG     P.BFCP(R5)      ;MAKE IT POSITIVE IT POSITIVE
023522  004737  023540          JSR     PC,CMPBUF       ;DO COMPARISON
023526  023534                  10$                     ;ERROR RETURN

023530  005720          5$:     TST     (R0)+           ;ADJUST R0 FOR RETURN
023532  000200                  RTS     R0              ;RETURN

023534  011000          10$:    MOV     (R0),R0         ;LOAD ERROR RETURN
023536  000200                  RTS     R0              ;RETURN
```

```
                                .SBTTL  COMPARE DATA BUFFER

                                ;;*****************************************************************
                                ;*
                                ;*      THE REGISTERS WILL BE USED AS FOLLOWS:
                                ;*
                                ;*      REGISTER                USE
                                ;*      --------                ---
                                ;*
                                ;*         R5           ADDRESS OF PARAMETER BLOCK
                                ;*         R4           BUFFER ADDRESS
                                ;*         R3           PATTERN ADDRESS
                                ;*         R2           SECTOR WORD COUNT
                                ;*         R1           WORDS ON SECTOR WRITTEN
                                ;*         R0           PATTERN COUNT
                                ;*
                                ;*CALL  JSR     PC,CMPBUF
                                ;*      <ADDRESS OF COMPARISON ERROR RETURN>
                                ;*      RETURN
                                ;*
                                ;*      (SP)    NUMBER OF SOFT DATA COMPARES
                                ;*      (SP+2)  WORD COUNT
                                ;*
                                ;;*****************************************************************

023540  010446          CMPBUF: MOV     R4,-(SP)        ;STORE R4
023542  010346                  MOV     R3,-(SP)        ;STORE R3
023544  010246                  MOV     R2,-(SP)        ;STORE R2
023546  010146                  MOV     R1,-(SP)        ;STORE R1
023550  010046                  MOV     R0,-(SP)        ;STORE R0
023552  016504  000214          MOV     P.ERHD(R5),R4   ;GET START ADDRESS
023556  016546  000222          MOV     P.BFCP(R5),-(SP) ;GET COMPARE COUNT
023562  016546  000220          MOV     P.CMLG(R5),-(SP) ;LOAD NUMBER OF COMPARISON
023566  001002                  BNE     10$             ;COMPARE IF COMPARE COUNT NOT ZERO
023570  000137  024024          JMP     40$             ;ELSE RETURN

023574  010465  000214  10$:    MOV     R4,P.ERHD(R5)   ;LOAD SECTOR HEAD IN CASE OF ERROR
023600  005002                  CLR     R2              ;CLEAR OFFSET FROM HEAD
023602  012400                  MOV     (R4)+,R0        ;STORE 1ST WORD (PATTERN INDEX)
023604  010003                  MOV     R0,R3
023606  042700  177760          BIC     #177760,R0      ;KEEP LOW 4 BITS
023612  042703  177417          BIC     #177417,R3      ;MASK CPU ID
023616  006203                  ASR     R3              ;SHIFT 4 BITS RIGHT
023620  006203                  ASR     R3
023622  006203                  ASR     R3
023624  006203                  ASR     R3
023626  020003                  CMP     R0,R3           ;CHECK FOR PATTERN MATCH
023630  001066                  BNE     30$             ;NO. REPORT ERROR
023632  005202                  INC     R2              ;INCREMENT OFFSET
023634  005366  000002          DEC     2(SP)           ;DECREMENT COMPARISON COUNT

023640  006300          13$:    ASL     R0              ;MULTIPLY INDEX BY 2
023642  016003  004602          MOV     PATTAB(R0),R3   ;STORE PATTERN ADDRESS
023646  126414  000001          CMPB    1(R4),(R4)      ;CHECK WORD COUNT
023652  001055                  BNE     30$             ;NO, REPORT ERROR
023654  012401                  MOV     (R4)+,R1        ;STORE WORDS IN SECTOR WRITTEN
```

```
5732  023656  042701  177400              BIC     #177400,R1      ;KEEP LOW BYTE
5733  023662  005202                      INC     R2              ;INCREMENT OFFSET VALUE
5734  023664  005366  000002              DEC     2(SP)           ;DECREMENT WORD COUNT
5735  023670  005765  000116              TST     P.SMPL(R5)      ;CHECK IF SAMPLED COMPARE
5736  023674  001467                      BEQ     50$             ;YES, START SAMPLED COMPARE SEQUENCE
5737  023676  005701                      TST     R1              ;CHECK IF SECTOR ALL ZEROES
5738  023700  001426                      BEQ     20$             ;YES GO TO ZERO FILL LOOP
5739  023702  012700  000040              MOV     #32.,R0         ;LOAD PATTERN COUNT
5740  023706  022324              15$:    CMP     (R3)+,(R4)+     ;CHECK PATTERN
5741  023710  001036                      BNE     30$             ;MISCOMPARE ERROR
5742  023712  005366  000002              DEC     2(SP)           ;DECREMENT WORD COUNT
5743  023716  001442                      BEQ     40$             ;NO MORE COMPARISONS
5744  023720  005316                      DEC     (SP)            ;DECREMENT NUMBER OF SOFT
5745                                                              ;  DATA COMPARE
5746  023722  001440                      BEQ     40$             ;NO MORE COMPARISONS
5747  023724  005202                      INC     R2              ;INCREMENT OFFSET
5748  023726  022702  000400              CMP     #256.,R2        ;CHECK IF COMPLETE SECTOR COMPARES
5749  023732  001720                      BEQ     10$             ;YES, START NEXT SECTOR
5750  023734  005301                      DEC     R1              ;DECREMENT NUMBER OF WORDS WRITTEN
5751  023736  001407                      BEQ     20$             ;IF ZERO, GO TO ZERO FILL LOOP
5752  023740  005300                      DEC     R0              ;DECREMENT PATTERN COUNT
5753  023742  001361                      BNE     15$             ;NEXT WORD
5754  023744  012700  000040              MOV     #32.,R0         ;REINITIALIZE PATTERN COUNT
5755  023750  162703  000100              SUB     #64.,R3         ;REINITIALIZE PATTERN ADDRESS
5756  023754  000754                      BR      15$             ;CHECK NEXT WORD
5757
5758  023756  005724              20$:    TST     (R4)+           ;CHECK ZERO FILL OF PATTERN
5759  023760  001012                      BNE     30$             ;MISCOMPARE ERROR
5760  023762  005366  000002              DEC     2(SP)           ;DECREMENT WORD COUNT
5761  023766  001416                      BEQ     40$             ;NO MORE COMPARISONS
5762  023770  005316                      DEC     (SP)            ;DECREMENT NUMBER OF SOFT
5763                                                              ;  DATA COMPARES
5764  023772  001414                      BEQ     40$             ;NO MORE COMPARISONS
5765  023774  005202                      INC     R2              ;INCREMENT OFFSET VALUE
5766  023776  022702  000400              CMP     #256.,R2        ;CHECK IF COMPLETE SECTOR COMPARED
5767  024002  001674                      BEQ     10$             ;YES, START NEXT SECTOR
5768  024004  000764                      BR      20$             ;COMPARE NEXT WORD ON
5769
5770  024006  110265  000217      30$:    MOVB    R2,P.ERAD(R5)   ;STORE OFFSET ADDRESS
5771  024012  022626                      CMP     (SP)+,(SP)+     ;ADJUST STACK
5772  024014  017666  000012  000012      MOV     @12(SP),12(SP)  ;LOAD ERROR RETURN
5773  024022  000404                      BR      45$             ;RETURN
5774
5775  024024  022626              40$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
5776  024026  062766  000002  000012      ADD     #2,12(SP)       ;ADJUST RETURN
5777
5778  024034  105065  000117      45$:    CLRB    P.ECMP(R5)      ;CLEAR ECC COMPARE FLAG
5779  024040  012600                      MOV     (SP)+,R0        ;RESTORE R0
5780  024042  012601                      MOV     (SP)+,R1        ;RESTORE R1
5781  024044  012602                      MOV     (SP)+,R2        ;RESTORE R2
5782  024046  012603                      MOV     (SP)+,R3        ;RESTORE R3
5783  024050  012604                      MOV     (SP)+,R4        ;RESTORE R4
5784  024052  000207                      RTS     PC              ;RETURN
5785
5786  024054  005701              50$:    TST     R1              ;CHECK IF ALL ZEROES
5787  024056  001432                      BEQ     52$             ;YES, TEST FOR ZEROES
```

```
5788  024060  021314                          CMP     (R3),(R4)       ;CHECK FIRST WORD OF PATERN
5789  024062  001351                          BNE     30$             ;MISCOMPARE, REPORT ERROR
5790  024064  026601  000002                  CMP     2(SP),R1        ;CHECK IF ENOUGH WORDS READ FOR
5791                                                                    ;  BOUNDARY COMPARISON
5792  024070  103443                          BLO     55$             ;NO, DO SAMPLE COMPARE
5793  024072  010146                          MOV     R1,-(SP)        ;SAVE SECTOR COUNT
5794  024074  005301                          DEC     R1              ;CALCULATE NUMBER OF WORDS FROM START
5795  024076  006301                          ASL     R1              ;CONVERT TO BYTES
5796  024100  060104                          ADD     R1,R4           ;CALCULATE MEMORY ADDRESS
5797  024102  042701  177700                  BIC     #177700,R1      ;GET PATTERN ADDRESS OFFSET
5798  024106  060103                          ADD     R1,R3           ;CALCULATE PATTERN ADDRESS
5799  024110  021324                          CMP     (R3),(R4)+      ;CHECK IF DATA WORD IS CORRECT
5800  024112  001403                          BEQ     51$             ;YES, CHECK ZERO FILL
5801  024114  005216                          INC     (SP)            ;CALCULATE OFFSET
5802  024116  012602                          MOV     (SP)+,R2        ;STORE OFFSET
5803  024120  000732                          BR      30$             ;REPORT ERROR
5804
5805  024122  012602                  51$:    MOV     (SP)+,R2        ;STORE OFFSET
5806  024124  062702  000002                  ADD     #2,R2           ;CALCULATE OFFSET FOR START OF ZERO FILL
5807  024130  022702  000377                  CMP     #377,R2         ;CHECK IF NO ZERO FILL
5808  024134  103421                          BLO     55$             ;YES, DO SAMPLE COMPARE
5809  024136  026602  000002                  CMP     2(SP),R2        ;CHECK WORD WAS READ
5810  024142  103416                          BLO     55$             ;NO, DO SAMPLE COMPARE
5811  024144  005714                  52$:    TST     (R4)            ;CHECK START OF ZERO FILL
5812  024146  001317                          BNE     30$             ;NO, REPORT ERROR
5813  024150  022766  000377  000002          CMP     #377,2(SP)      ;CHECK IF LAST WORD READ
5814  024156  101010                          BHI     55$             ;NO, DO SAMPLE COMPARE
5815  024160  012702  000377                  MOV     #377,R2         ;LOAD OFFSET COUNT
5816  024164  016504  000214                  MOV     P.ERHD(R5),R4   ;LOAD START OF SECTOR
5817  024170  062704  000776                  ADD     #776,R4         ;CALCULATE END OF SECTOR
5818  024174  005714                          TST     (R4)            ;CHECK LAST WORD OF SECTOR
5819  024176  001303                          BNE     30$             ;REPORT ERROR
5820
5821  024200  016504  000214          55$:    MOV     P.ERHD(R5),R4   ;SET START OF SECTOR
5822  024204  012400                          MOV     (R4)+,R0        ;GET PATTERN INDEX
5823  024206  042700  177760                  BIC     #177760,R0      ;MASK OFF UNNECESSARY BIT
5824  024212  006300                          ASL     R0              ;MULTIPLY BY 2
5825  024214  016003  004602                  MOV     PATTAB(R0),R3   ;GET PATTERN ADDRESS
5826  024220  012401                          MOV     (R4)+,R1        ;GET SECTOR WORD COUNT
5827  024222  042701  177400                  BIC     #177400,R1      ;MASK OFF UNNECESSARY BITS
5828  024226  012702  000002                  MOV     #2,R2           ;INITIALIZE OFFSET
5829  024232  012700  000007                  MOV     #7,R0           ;LOAD NUMBER OF SAMPLES
5830  024236  162701  000043          57$:    SUB     #35.,R1         ;SUBTRACT 35 FROM SECTOR WORD COUNT
5831  024242  101430                          BLOS    60$             ;CHECK IF ZERO FILL
5832  024244  162766  000043  000002          SUB     #35.,2(SP)      ;SUBTRACT 35 FROM WORD COUNT
5833  024252  101664                          BLOS    40$             ;CHECK IF FINISHED
5834  024254  062702  000043                  ADD     #35.,R2         ;CALCULATE OFFSET
5835  024260  062704  000106                  ADD     #106,R4         ;DETERMINE ADDRESS OF COMPARE
5836  024264  062703  000006                  ADD     #6,R3           ;DETERMINE ADDRESS OF PATTERN
5837  024270  021413                          CMP     (R4),(R3)       ;CHECK IF DATA CORRECT
5838  024272  001245                          BNE     30$             ;NO, REPORT ERROR
5839  024274  005316                          DEC     (SP)            ;DECREMENT NUMBER OF COMPARES
5840  024276  001652                          BEQ     40$             ;CHECK IF FINISHED
5841  024300  005300                          DEC     R0              ;CHECK IF IF TIME FOR NEXT SECTOR
5842  024302  001355                          BNE     57$             ;NO, CHECK NEXT SAMPLE
5843  024304  162766  000011  000002          SUB     #11,2(SP)       ;ADJUST WORD COUNT
```

```
 5844  024312  101644                          BLOS    40$             ;CHECK IF FINISHED
 5845  024314  062704  000022                  ADD     #22,R4          ;CALCULATE START OF NEXT SECTOR
 5846  024320  000137  023574                  JMP     10$             ;GO SERVICE NEXT SECTOR
 5847
 5848  024324  162766  000043  000002  60$:    SUB     #35.,2(SP)      ;SUBTRACT 35 FROM WORD COUNT
 5849  024332  101634                          BLOS    40$             ;CHECK IF FINISHED
 5850  024334  062702  000043                  ADD     #35.,R2         ;DETERMINE OFFSET
 5851  024340  062704  000106                  ADD     #106,R4         ;DETERMINE ADDRESS OF COMPARE
 5852  024344  005714                          TST     (R4)            ;CHECK IF DATA CORRECT
 5853  024346  001217                          BNE     30$             ;NO, REPORT ERROR
 5854  024350  005316                          DEC     (SP)            ;DECREMENT NUMBER OF COMPARES
 5855  024352  001624                          BEQ     40$             ;BRANCH IF FINISHED
 5856  024354  005300                          DEC     R0              ;CHECK IF SAMPLE COUNT EXHAUSTED
 5857  024356  001362                          BNE     60$             ;NO, CHECK NEXT SAMPLE
 5858  024360  162766  000011  000002          SUB     #11,2(SP)       ;ADJUST WORD COUNT
 5859  024366  101616                          BLOS    40$             ;CHECK IF FINISHED
 5860  024370  062704  000022                  ADD     #22,R4          ;CALCULATE START OF NEXT SECTOR
 5861  024374  000137  023574                  JMP     10$             ;GO SERVICE NEXT SECTOR
```

```
                                        .SBTTL  NORMAL COMMAND RETURN

                                        ;;*****************************************************************
                                        ;*
                                        ;*      THIS POUTINE 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
                                        ;*
                                        ;;*****************************************************************

024400  004737  047542          NORMAL: JSR     PC,I.CSTS       ;GATHER CONTROLLER STATUS
024404  032765  024000  000016          BIT     #SPAR!CTO,P.CS1(R5) ;CHECK IF DRIVE BUS PARITY OR CONTROLLER TIME OUT
024412  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
024414  032765  177400  000020          BIT     #UFE!MDS!PGE!NEM!NED!UPE!WCE!DLT,P.CS2(R5)
024422  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
024424  032765  131761  000034          BIT     #ILC!FMTE!DTYE!ECH!BSE!HVRC!COE!DTE!OPI!DCK,P.ER(R5)
024432  001405                        BEQ     3$              ;NO, CHECK FOR UNREPORT DRIVE ERRORS
024434  052737  000001  001564  2$.     BIS     #BIT0,ERCONT    ;SET CONTROLLER ERROR NOT FLAGGED
024442  000137  027342                JMP     CNTRL1          ;REPORT CONTROLLER ERROR
```

```
5918
5919                                    ;       CHECK THE FOLLOWING ERROR BITS IN RKER(UNREPORTED DRIVE ERROR)
5920                                    ;               SEEK INCOMPLETE
5921                                    ;               NON-EXECUTABLE DRIVE FUNCTION
5922                                    ;               DRIVE DETECTED DRIVE BUS PARITY ERROR
5923                                    ;               INVALID DISK ADDRESS
5924                                    ;               WRITE LOCK ERROR
5925                                    ;               UNSAFE
5926  024446  032765  046016  000034  3$:     BIT     #SKI!NXF!DRPAR!IDAE!WLE!UNS,P.ER(R5)
5927  024454  001004                          BNE     4$              ;YES,SET UNREPORTED DRIVE ERROR
5928
5929                                    ;       CHECK THE FOLLOWING ERROR BITS IN RKDS
5930                                    ;               AC LOW
5931                                    ;               SPEED LOSS
5932                                    ;               DRIVE OFF TRACK
5933  024456  032765  000070  000036          BIT     #ACLO!SPDLSS!DROT,P.DS(R5)
5934  024464  001405                          BEQ     5$              ;NO, CHECK IF SPECIAL SEQUENCE
5935  024466  052765  000010  000212  4$:     BIS     #BIT3,P.ERR(R5) ;SET DRIVE ERROR NOT REPORTED
5936  024474  000137  025470                  JMP     ABNORM          ;PROCESS ERROR
5937
5938  024500  032765  000001  000014  5$:     BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
5939  024506  001005                          BNE     6$              ;YES, CHECK IF SPECIAL SEQUENCE
5940  024510  052765  000200  000212          BIS     #BIT7,P.ERR(R5) ;SET UNSOLITATED ATTENTION
5941  024516  000137  025470                  JMP     ABNORM          ;PROCESS ERROR
5942
5943  024522  005737  001564          6$:     TST     ERCONT          ;CHECK IF CONTROLLER ERROR HAS OCCURRED
5944  024526  100017                          BPL     11$             ;NO, CONTINUE NORMAL PROCESSING
5945  024530  122765  000121  000001          CMPB    #RDDATA,P.CMND(R5) ;CHECK IF DATA TRANSFER COMMAND
5946  024536  101007                          BHI     7$              ;NO, REISSUE COMMAND
5947  024540  132765  000040  000001          BITB    #BIT5,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
5948  024546  001003                          BNE     7$              ;YES, REISSUE COMMAND
5949  024550  004037  025210                  JSR     R0,CHKADD       ;CHECK BUS AND DISK ADDRESS
5950  024554  024564                          10$                     ;ERROR RETURN
5951  024556  004037  051564          7$:     JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
5952  024562  001324                          CINITQ                  ;  COMMAND INITIATION QUEUE
5953  024564  000207                  10$:    RTS     PC              ;RETURN
5954
5955  024566  005765  000150          11$:    TST     P.DSTT(R5)      ;CHECK IF RETRY SEQUENCE
5956  024572  001417                          BEQ     32$             ;NO, CHECK IF DRIVE IS BEING ASSIGNED
5957  024574  004737  040440                  JSR     PC,ERCVY1       ;GO THROUGH ERROR RECOVERY
5958  024600  005737  001566          20$:    TST     ERRPRO          ;CHECK IF ERROR PROCESSING COMPLETE
5959  024604  001367                          BNE     10$             ;NO, RETURN
5960  024606  004037  051644                  JSR     R0,Q.POP        ;GET NEXT ERROR TO BE PROCESSED
5961  024612  001330                          ERRPRQ
5962  024614  012605                          MOV     (SP)+,R5        ;LOAD ADDRESS OF PARAMETER BLOCK FOR
5963                                                                  ;  NEXT ERROR
5964  024616  001762                          BEQ     10$             ;IF NO MORE ERRORS, RETURN
5965  024620  010537  001566                  MOV     R5,ERRPRO       ;INDICATE THAT ERROR IS BEING PROCESSED
5966  024624  004737  032160                  JSR     PC,ERCVY        ;GO THROUGH ERROR RECOVERY
5967  024630  000763                          BR      20$             ;CHECK IF ERROR PROCESSING COMPLETE
5968
5969  024632  105765  000210          32$:    TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
5970  024636  001402                          BEQ     33$             ;NO, CHECK IF SEEK OR RELEASE
5971  024640  000137  013662                  JMP     ASNORM          ;GO CONTINUE ASSIGNMENT
5972
5973  024644  122765  000117  000001  33$:    CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK TO PREVIOUS POSITION
```

```
                                                        ;  (SWITCH 14 OPTION)
024652  001416                      BEQ     34$           ;YES, RELEASE DRIVE
024654  122765  000140  000001      CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE COMMAND
024662  001021                      BNE     35$           ;NO, CHECK ADDRESS AND WORD COUNT
024664  112737  177777  001506      MOVB    #-1,I.ISRL    ;INDICATE THAT AN INTERRUPT HAS BEEN ISSUED
024672  012762  000300  000000      MOV     #INTR,RKCS1(R2) ;FORCE AN INTERRUPT SCAN DRIVE ATTENTIONS
024700  004037  051564              JSR     R0,Q.PUSH     ;PUT PARAMTER BLOCK ADDRESS ON
024704  001314                      AVAILQ                ;  DRIVE AVAILIABLE QUEUE
024706  000207                      RTS     PC            ;RETURN

024710  112765  000140  000001  34$:MOVB    #RELEAS,P.CMND(R5) ;PUT RELEASE COMMAND IN PARAMETER BLOCK
024716  004037  051564              JSR     R0,Q.PUSH     ;ENQUEUE PARAMETER BLOCK IN
024722  001324                      CINITQ                ;  COMMAND INITIATION QUEUE
024724  000207                      RTS     PC            ;RETURN

024726  004037  025210          35$:JSR     R0,CHKADD     ;CHECK WORD COUNT, BUS ADDRESS,
                                                        ;  SECTOR, TRACK, AND CYLINDER
024732  024564                      10$                   ;ERROR RETURN ADDRESS
024734  112737  177777  001506      MOVB    #-1,I.ISRL    ;INDICATE THAT AN INTERRUPT HAS BEEN ISSUED
024742  012762  000300  000000      MOV     #INTR,RKCS1(R2) ;GO SCAN FOR DRIVE INTERRUPTS
024750  005037  177776              CLR     PS            ;ALLOW RK06 INTERRUPTS
024754  122765  000121  000001      CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ DATA
024762  001004                      BNE     NORM1         ;NO, DO NOT COMPARE BUFFER
024764  004037  023462              JSR     R0,BUFER1     ;CHECK DATA READ
024770  025602                      DATERR                ;ERROR RETURN
024772  000417                      BR      NORM2         ;RELEASE BUFFER

024774  122765  000131  000001 NORM1:CMPB   #WRTCHK,P.CMND(R5) ;CHECK IF WRITE COMMAND
025002  001013                      BNE     NORM2         ;NO, RELEASE BUFFER
025004  032765  000200  000014      BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK ISSUED
025012  001407                      BEQ     NORM2         ;YES, RELEASE BUFFER
025014  042765  000200  000014      BIC     #W.WCK,P.PRST(R5) ;CLEAR WRITE FLAG
025022  004037  051564              JSR     R0,Q.PUSH     ;PUT PARAMETER BLOCK IN
025026  001324                      CINITQ                ;  COMMAND INITIATION QUEUE
025030  000207                      RTS     PC            ;RETURN

025032  013737  001452  177776 NORM2:MOV    RKPRI,PS      ;LOCK OUT RK06 INTERRUPTS
025040  004737  023020              JSR     PC,BUFREL     ;RELEASE BUFFER
025044  005037  177776              CLR     PS            ;ALLOW RK06 INTERRUPTS
025050  032777  040040  154062      BIT     #SW5!SW14,@SWR ;CHECK IF INHIBIT AUTOMATIC DESELECT
025056  001043                      BNE     33$           ;YES, PUT PARAMETER BLOCK IN
                                                        ;  AVAILIABLE QUEUE
025060  026565  000110  000154      CMP     P.MXCD+2(R5),P.NODR+2(R5) ;CHECK IF MAXIMUM NUMBER
                                                        ;  OF COMMAND HAVE BEEN ISSUED
025066  101010                      BHI     29$           ;NO, CHECK IF MAXIMUM NUMBER OF
                                                        ;  OF WORDS HAVE BEEN TRANSFERRED
025070  103404                      BLO     28$           ;YES, DROP DRIVE
025072  026565  000106  000152      CMP     P.MXCD(R5),P.NODR(R5) ;CHECK IF MAXIMUM NUMBER
                                                        ;  OF COMMANDS HAVE BEEN ISSUED
025100  103003                      BHIS    29$           ;NO, CHECK IF MAXIMUM NUMBER OF
                                                        ;  HAVE BEEN TRANSFERRED
025102  004737  013162          28$:JSR     PC,DROP       ;DROP DRIVE
025106  000435                      BR      35$           ;ALLOCATE BUFFER


025110  016546  000166          29$:MOV     P.NRD+2(R5),-(SP) ;STORE NUMBER OF WORDS READ ON STACK
```

```
6030  025114  016546  000170                  MOV     P.NRD+4(R5),-(SP)
6031  025120  066566  000160  000002          ADD     P.NWRT+2(R5),2(SP) ;ADD NUMBER OF WORDS WRITTEN
6032  025126  005516                          ADC     (SP)
6033  025130  066516  000162                  ADD     P.NWRT+4(R5),(SP)
6034  025134  103410                          BCS     30$             ;DROP DRIVE IF OVERFLOW
6035  025136  026526  000114                  CMP     P.MXWT+2(R5),(SP)+ ;CHECK IF MAXIMUM NUMBER OF WORDS
6036                                                                  ;  HAVE BEEN TRANSFERRED
6037  025142  101010                          BHI     32$             ;NO, PUT PARAMETER BLOCK IN
6038                                                                  ;  AVAILIABLE QUEUE
6039  025144  103405                          BLO     31$             ;YES, DROP DRIVE
6040  025146  026526  000112                  CMP     P.MXWT(R5),(SP)+ ;CHECK IF MAXIMUM NUMBER OF WORDS
6041                                                                  ;  HAVE BEEN TRANSFERRED
6042  025152  103753                          BLO     28$             ;YES, DROP DRIVE
6043  025154  000404                          BR      33$             ;NO, PUT PARAMETER BLOCK IN
6044                                                                  ;  AVAILABLE QUEUE
6045
6046  025156  005726                  30$:    TST     (SP)+           ;ADJUST STACK
6047  025160  005726                  31$:    TST     (SP)+           ;ADJUST STACK
6048  025162  000747                          BR      28$             ;DROP DRIVE
6049
6050  025164  005726                  32$:    TST     (SP)+           ;ADJUST STACK
6051  025166  112765  000140  000001  33$:    MOVB    #RELEAS,P.CMND(R5) ;PUT RELEASE COMMAND IN PARAMETER BLOCK
6052  025174  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
6053  025200  001324                          CINITQ                  ;  COMMAND INITIATION QUEUE
6054
6055  025202  004737  022520          35$:    JSR     PC,GETBUF       ;GET NEW BUFFER
6056  025206  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  CHECK BUS ADDRESS, WORD COUNT, SECTOR, TRACK, AND CYLINDER
                                CHKADD: TST     P.WCR(R5)         ;CHECK IF WORD COUNT ZERO
                                        BEQ     6$                ;YES, CHECK BUS ADDRESS
                                        BIS     #BIT1,ERCONT      ;SET WORD COUNT NOT EQUAL ZERO
                                        BR      24$               ;REPORT ERROR

                                6$:     BIT     #BA16!BA17,P.CS1(R5) ;CHECK IF HIGH BUS ADDRESS BITS SET
                                        BNE     10$               ;YES, REPORT ERROR
                                        MOV     P.WC(R5),-(SP)    ;STORE WORD COUNT
                                        NEG     (SP)              ;MAKE IT POSITIVE
                                        ASL     (SP)              ;DETERMINE NUMBER OF BITS
                                        ADD     P.BALO(R5),(SP)   ;DETERMINE EXPECTED BUS ADDRESS
                                        CMP     P.BAR(R5),(SP)+   ;CHECK BUS ADDRESS
                                        BEQ     17$               ;BUS ADDRESS CORRECT--CHECK CYLINDER,
                                                                  ;TRACK, AND SECTOR
                                10$:    BIS     #BIT2,ERCONT      ;SET BUS ADDRESS INCORRECT
                                        BR      24$               ;REPORT ERROR

                                17$:    MOV     P.WC(R5),-(SP)    ;STORE WORD COUNT
                                        NEG     (SP)              ;MAKE IT POSITIVE
                                        TSTB    (SP)              ;CHECK IF EVEN NUMBER OF SECTORS
                                        BEQ     18$               ;YES, DO COMPARISON
                                        INCB    1(SP)             ;INCREMENT SECTOR COUNT
                                18$:    CLR     -(SP)             ;MAKE ROOM ON STACK
                                        MOVB    3(SP),(SP)        ;STORE NUMBER OF SECTORS TRANSFERRED
                                        CLR     2(SP)             ;CLEAR LOCATION ON STACK
                                        MOVB    P.SECT(R5),2(SP)  ;STORE STARTING SECTOR
                                        ADD     2(SP),(SP)        ;DETERMINE FINAL SECTOR ADDRESS
                                        CLR     2(SP)             ;CLEAR NUMBER OF TRACKS TRANSFERRED

                                19$:    CMP     #22.,(SP)         ;CHECK FOR SECTOR OVERFLOW
                                        BHI     20$               ;NO, CHECK IF SECTOR CORRECT
                                        SUB     #22.,(SP)         ;DECREMENT SECTOR COUNT BY 22 AND
                                        INC     2(SP)             ;  INCREMENT TRACKS TRANSFERRED
                                        BR      19$               ;CHECK FOR SECTOR OVERFLOW

                                20$:    CMPB    P.DTS(R5),(SP)+   ;CHECK IF FINAL SECTOR CORRECT
                                        BNE     23$               ;NO, REPORT ERROR
                                        CLR     -(SP)             ;MAKE ROOM FOR TRACKS TRANSFERRED
                                        MOVB    P.TRCK(R5),(SP)   ;STORE STARTING TRACK
                                        ADD     2(SP),(SP)        ;DETERMINE FINAL TRACK ADDRESS
                                        CLR     2(SP)             ;CLEAR FINAL CYLINDER

                                21$:    CMPB    #3,(SP)           ;CHECK FOR TRACK OVERFLOW
                                        BHI     22$               ;NO, CHECK FINAL TRACK
                                        SUB     #3,(SP)           ;DECREMENT TRACK COUNT BY 3 AND
                                        INC     2(SP)             ;  INCREMENT CYLINDER COUNT
                                        BR      21$               ;CHECK FOR TRACK OVERFLOW

                                22$:    CMPB    P.DTS+1(R5),(SP)+ ;CHECK IF FINAL TRACK CORRECT
                                        BNE     23$               ;TRACK INCORRECT ERROR
                                        ADD     P.CYLN(R5),(SP)   ;CALCULATE FINAL CYLINDER
                                        CMP     P.DCYL(R5),(SP)   ;CHECK IF CYLINDER CORRECT
                                        BEQ     25$               ;CORRECT GO TO NORMAL RETURN
                                23$:    TST     (SP)+             ;ADJUST STACK
```

```
 6113  025444  052737  000010  001564          BIS     #BIT3,ERCONT    ;SET INCORRECT SECTOR, TRACK, OR CYLINDER
 6114  025452  004737  027342          24$:    JSR     PC,CNTRL1       ;REPORT ERROR
 6115  025456  011000                          MOV     (R0),R0         ;LOAD R0 FOR ERROR RETURN
 6116  025460  000200                          RTS     R0              ;RETURN
 6117
 6118  025462  005726                  25$:    TST     (SP)+           ;ADJUST STACK
 6119  025464  005720                          TST     (R0)+           ;ADJUST R0
 6120  025466  000200                          RTS     R0              ;NORMAL RETURN
```

```
                                        .SBTTL  ABNORMAL TERMINATION

025470  004737  052004          ABNORM: JSR     PC,Q.RMOV       ;REMOVE PARAMETER BLOCK FROM COMMAND
                                        ;  INITIATION QUEUE
025474  032765  000100  000014          BIT     #CMDTO,P.PRST(R5) ;CHECK IF COMMAND TIME OUT
025502  001402                          BEQ     3$              ;NO, PROCESS ABNORMAL TERMINATION
025504  000137  025674                  JMP     TIMOUT          ;JUMP TO TIME OUT ROUTINE

025510  005737  001564          3$:     TST     ERCONT          ;CHECK IF CONTROLLER ERROR HAS OCCURRED
025514  100010                          BPL     10$             ;NO, CONTINUE ERROR PROCESSING
025516  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5) ;CHECK FOR UNSOLICATED INTERRUPT
025524  001403                          BEQ     5$              ;YES, DO NOT ENQUEUE PARAMETER BLOCK
025526  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
025532  001324                          CINITQ                  ;  COMMAND INITIATION QUEUE
025534  000207                  5$:     RTS     PC              ;RETURN

025536  032765  000001  000014  10$:    BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
025544  001004                          BNE     ABN000          ;YES, CHECK IF DRIVE BEING ASSIGNED
025546  052765  000200  000212          BIS     #BIT7,P.ERR(R5) ;INDICATE UNSOLICATED INTERRUPT
025554  000420                          BR      ABN001          ;CHECK IF ERROR IS BEING PROCESSED

025556  005765  000150          ABN000: TST     P.DSTT(R5)      ;CHECK IF CURRENTLY UNDERGOING
                                        ;  ERROR PROCESSING
025562  001415                          BEQ     ABN001          ;NO, CHECK IF ERROR PRESENTLY BEING PROCESSED
025564  100402                          BMI     13$             ;ERROR ENQUEUED
025566  000137  043236                  JMP     ERCVY2          ;ERROR RECOVERY ROUTINE

025572  052765  000100  000212  13$:    BIS     #BIT6,P.ERR(R5) ;SET ERROR WHILE ENQUEUED IN ERROR QUEUE
025600  000207                          RTS     PC              ;RETURN

025602  013737  001452  177776  DATERR: MOV     RKPRI,PS        ;LOCK RK06 INTERRUPTS
025610  052765  000004  000212          BIS     #BIT2,P.ERR(R5) ;SET SOFTWARE DETECTED DATA ERROR

025616  012765  100200  000150  ABN001: MOV     #BIT15!BIT7,P.DSTT(R5) ;SET ERROR ENQUEUED AND FIRST ERROR
025624  005737  001566                  TST     ERRPRO          ;CHECK IF ERROR IS BEING PROCESSED
025630  001015                          BNE     5$              ;YES, GO ENQUEUE ERROR
025632  010537  001566          2$:     MOV     R5,ERRPRO       ;LOAD ERROR BEING PROCESSED
025636  004737  032160                  JSR     PC,ERCVY        ;START ERROR RECOVERY
025642  005737  001566                  TST     ERRPRO          ;CHECK IF ERROR PROCESSING THROUGH
025646  001005                          BNE     3$              ;NO, RETURN
025650  004037  051644                  JSR     R0,Q.POP        ;GET NEXT ERROR ENQUEUED
025654  001330                          ERRPRQ
025656  012605                          MOV     (SP)+,R5        ;LOAD R5 FOR ERROR PROCESSING
025660  001364                          BNE     2$              ;IF NOT ZERO PROCESS NEXT ERROR
025662  000207                  3$:     RTS     PC              ;RETURN

025664  004037  051564          5$:     JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON
025670  001330                          ERRPRQ                  ;  ERROR PROCESSING QUEUE
025672  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  COMMAND TIME OUT ROUTINE

025674 016246 000010            TIMOUT: MOV     RKCS2(R2),-(SP) ;GET CS2 FOR DRIVE NUMBER
025700 042716 177770                    BIC     #^C<DRVMSK>,(SP) ;KEEP DRIVE NUMBER
025704 122665 000000                    CMPB    (SP)+,P.DRVN(R5) ;CHECK IF TIME OUT IF FOR
                                                                 ;  CURRENTLY SELECTED DRIVE
025710 001014                           BNE     5$              ;NO, TEST IF LOOPING ON CONTROLLER ERROR
025712 016237 000000 001410             MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REGISTER 1
025720 032737 000001 001410             BIT     #GO,T.CS1       ;CHECK IF GO IS STILL SET
025726 001405                           BEQ     5$              ;NO, TEST IF LOOPING ON CONTROLLER ERROR
025730 052737 040000 001462             BIS     #E.CMTO,E.CONT  ;SET CONTROLLER TIMED OUT ON COMMAND
025736 000137 026524                    JMP     CONTRL          ;GO REPORT CONTROLLER ERROR

025742 005737 001564            5$:     TST     ERCONT          ;CHECK IF CYCLING ON CONTROLLER ERROR
025746 100004                           BPL     10$             ;NO, CHECK IF DRIVE IS CURRENTLY
                                                                 ;  SEIZED BY OTHER PORT
025750 004037 051564                    JSR     R0,Q.PUSH       ;REQUEUE COMMAND IN COMMAND
025754 001324                           CINITQ                  ;  INITIATION QUEUE
025756 000207                           RTS     PC              ;RETURN

025760 032765 010000 000014     10$:    BIT     #DRVSZD,P.PRST(R5) ;CHECK IF DRIVE WAS SEIZED
025766 001404                           BEQ     12$             ;NO, CONTINUE
025770 052765 000020 000212             BIS     #BIT4,P.ERR(R5) ;SET TIME OUT BECAUSE DRIVE SEIZED
025776 000667                           BR      ABN000          ;GO INITIATE ERROR PROCESSING

026000 032765 020000 000014     12$:    BIT     #E.UNLD,P.PRST(R5) ;CHECK IF WAITING FOR HEADS TO RELOAD
026006 001404                           BEQ     15$             ;NO, SET TIME OUT FOR POSITIONING
026010 052765 000400 000212             BIS     #BIT8,P.ERR(R5) ;SET TIME OUT IN RELOADING HEADS
026016 000657                           BR      ABN000          ;GO INITIATE ERROR PROCESSING

026020 052765 000040 000212     15$:    BIS     #BIT5,P.ERR(R5) ;SET TIME OUT DURING POSITIONING
026026 000653                           BR      ABN000          ;GO INITIATE ERROR PROCESSING
```

```
 6202
 6203
 6204  026030  010346                  .SBTTL  PRINT RK06 UNIBUS REGISTERS
 6205  026032  010146
 6206  026034  012703  001410  PRIREG: MOV     R3,-(SP)        ;STORE R3 ON STACK
 6207  026040  012701  001430          MOV     R1,-(SP)        ;STORE R1 ON STACK
 6208  026044  104401  065006          MOV     #T.CS1,R3       ;LOAD BEGINNING OF REGISTER STORAGE
 6209  026050  004737  026310          MOV     #T.DS,R1        ;LOAD END OF REGISTER STORAGE
 6210  026054  012701  001444          TYPE    ,ERR300         ;PRINT REGISTER NAMES
 6211  026060  104401  065105          JSR     PC,PRSTAT       ;PRINT REGISTER CONTENTS
 6212  026064  004737  026310          MOV     #T.DB,R1        ;LOAD END OF REGISTER STORAGE
 6213  026070  012601                  TYPE    ,ERR301         ;PRINT REGISTER NAMES
 6214  026072  012603                  JSR     PC,PRSTAT       ;PRINT REGISTER CONTENTS
 6215  026074  000207                  MOV     (SP)+,R1        ;RESTORE R1
 6216                                  MOV     (SP)+,R3        ;RESTORE R3
 6217                                  RTS     PC              ;RETURN
 6218
 6219  026076  010346                  .SBTTL  PRINT DRIVE STATUS
 6220  026100  104401  065165
 6221  026104  016546  000036  PRDSTT: MOV     R3,-(SP)        ;STORE R3 ON STACK
 6222  026110  004737  053224          TYPE    ,ERR302         ;PRINT REGISTER NAMES
 6223  026114  104401  065174          MOV     P.DS(R5),-(SP)  ;CONVERT DRIVE STATUS REG TO OCTAL
 6224  026120  010503                  JSR     PC,BINOCT
 6225  026122  062703  000040          TYPE    ,ERR303         ;PRINT HEADER
 6226  026126  012346                  MOV     R5,R3           ;DETERMINE START OF PRINT OUT
 6227  026130  004737  053224          ADD     #P.A00,R3
 6228  026134  104401  061126          MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
 6229  026140  012346                  JSR     PC,BINOCT
 6230  026142  004737  053224          TYPE    ,BLANKS
 6231  026146  104401  065235          MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
 6232  026152  012346                  JSR     PC,BINOCT
 6233  026154  004737  053224          TYPE    ,ERR304
 6234  026160  104401  061126          MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
 6235  026164  012346                  JSR     PC,BINOCT
 6236  026166  004737  053224          TYPE    ,BLANKS
 6237  026172  104401  065250          MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
 6238  026176  012346                  JSR     PC,BINOCT
 6239  026200  004737  053224          TYPE    ,ERR305
 6240  026204  104401  061126          MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
 6241  026210  012346                  JSR     PC,BINOCT
 6242  026212  004737  053224          TYPE    ,BLANKS
 6243  026216  104401  065263          MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
 6244  026222  012346                  JSR     PC,BINOCT
 6245  026224  004737  053224          TYPE    ,ERR306
 6246  026230  104401  061126          MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
 6247  026234  012346                  JSR     PC,BINOCT
 6248  026236  004737  053224          TYPE    ,BLANKS
 6249  026242  104401  001165          MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
 6250  026246  012603                  JSR     PC,BINOCT
 6251  026250  000207                  TYPE    ,$CRLF
                                       MOV     (SP)+,R3        ;RESTORE R3
                                       RTS     PC              ;RETURN
```

```
                                        .SBTTL  PRINT CONTROLLER STATUS
                                        
                                        PRISTT: MOV     R3,-(SP)        ;STORE R3 ON STACK
                                                MOV     R1,-(SP)        ;STORE R1 ON STACK
                                                MOV     R5,R3           ;STORE PARAMETER BLOCK ADDRESS
                                                MOV     R5,R1           ;  FOR INDEX CACULATIONS
                                                ADD     #P.CS1,R3       ;CALCULATE BEGINNING OF OUTPUT OCTAL
                                                ADD     #P.DS,R1        ;CALCULATE END OF OUTPUT OCTAL
                                                TYPE    ,ERR300         ;PRINT REGISTER NAMES
                                                JSR     PC,PRSTAT       ;PRINT REGISTER CONTENTS
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                MOV     (SP)+,R3        ;RESTORE R3
                                                RTS     PC              ;RETURN
                                        
                                        .SBTTL  PRINT REGISTER CONTENTS
                                        
                                        PRSTAT: MOV     (R3)+,-(SP)     ;SAVE REGISTER VALUE FOR TYPE OUT
                                                JSR     PC,BINOCT       ;CONVERT TO OCTAL
                                                CMP     R1,R3           ;CHECK IF FINISHED
                                                BEQ     10$             ;YES, RETURN
                                                TYPE    ,BLANKS         ;TYPE 2 BLANKS
                                                BR      PRSTAT          ;TYPE NEXT REGISTER
                                        
                                        10$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
                                                RTS     PC              ;RETURN
```

```
                                .SBTTL  ECC CORRECTION ROUTINE

                        ECCCOR: MOV     R4,-(SP)        ;STORE R4 ON STACK
                                MOV     R3,-(SP)        ;STORE R3 ON STACK
                                MOV     R1,-(SP)        ;STORE R1 ON STACK
                                MOV     R0,-(SP)        ;STORE R0 ON STACK
                                MOV     RTYBA,R3        ;LOAD BEGINNING OF SECTOR ADDRESS
                                MOV     RTYWC,R4        ;LOAD WORDS READ FROM SECTOR
                                ASL     R4              ;MAKE IT BYTES
                                MOV     R3,-(SP)        ;STORE ADDRESS OF START OF SECTOR
                                ADD     R4,(SP)         ;CALCULATE END OF SECTOR
                                MOV     P.EPOS(R5),R1   ;SAVE POSITION INFORMATION
                                DEC     R1              ;DECREMENT BIT COUNT BY ONE TO GET
                                                        ;  BIT ADDRESS
                                MOV     R1,R0
                                ASR     R1              ;DETERMINE WORDS THAT ARE BAD
                                ASR     R1              ;  FOR DATA READ (SHIFT 3 BITS
                                ASR     R1              ;  RIGHT)
                                BIC     #1,R1           ;CLEAR BYTE INDICATOR
                                CMP     R1,R4           ;CHECK IF ERROR WITHIN TRANSFER
                                BHIS    10$             ;NO, RETURN
                                ADD     R1,R3           ;DETERMINE ADDRESS OF ERROR
                                MOV     P.EPAT(R5),ECCPAT ;LOAD PATTERN
                                CLR     ECCPAT+2
                                BIC     #177760,R0      ;DETERMINE BIT POSITION FOR START OF ECC
                                                        ;  CORRECTION
                                BEQ     5$              ;CORRECTION STARTS ON
                                                        ;  WORD BOUNDARY

                        3$:     ASL     ECCPAT          ;SHIFT PATTERN 1 BIT LEFT
                                ROL     ECCPAT+2
                                DEC     R0              ;DECREMENT COUNT
                                BNE     3$              ;CHECK IF IN POSITION

                        5$:     MOV     (R3),R0         ;CORRECT FIRST WORD WITH EXCLUSIVE
                                MOV     ECCPAT,R1       ;  OF BITS IN ERROR
                                BIC     ECCPAT,(R3)
                                BIC     R0,R1
                                BIS     R1,(R3)+
                                CMP     (SP),R3         ;CHECK IF SECOND WORD IN TRANSFER
                                BEQ     10$             ;NO, RETURN
                                MOV     (R3),R0         ;CORRECT SECOND WORD WITH EXCLUSIVE
                                MOV     ECCPAT+2,R1     ;  OF BITS IN ERROR
                                BIC     ECCPAT+2,(R3)
                                BIC     R0,R1
                                B S     R1,(R3)

                        10$:    TST     (SP)+           ;ADJUST STACK
                                MOV     (SP)+,R0        ;RESTORE R0
                                MOV     (SP)+,R1        ;RESTORE R1
                                MOV     (SP)+,R3        ;RESTORE R3
                                MOV     (SP)+,R4        ;RESTORE R4
                                RTS     PC              ;RETURN
```

```
                                        .SBTTL  CONTROLLER ERROR DETECTED BY RK06 DRIVER

026524 010446                   CONTRL: MOV     R4,-(SP)        ;STORE R4 ON STACK
026526 004737 047414                    JSR     PC,I.STOR       ;GET CONTROLLER STATUS
026532 032777 002000 152400             BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
026540 001402                           BEQ     1$              ;NO, CONTINUE
026542 104401 001160                    TYPE    ,$BELL          ;RING BELL
026546 013704 001462            1$:     MOV     E.CONT,R4       ;STORE CONTROLLER INDICATORS
026552 032704 100000                    BIT     #E.MDS,R4       ;CHECK IF DUPLICATE DRIVE SELECT
026556 001403                           BEQ     2$              ;NO, CHECK FOR OTHER ERRORS
026560 104401 061400                    TYPE    ,ERR006         ;TYPE "MULTIPLE DRIVE SELECT"
026564 000501                           BR      10$             ;DISPLAY REGISTERS

026566 032704 000001            2$:     BIT     #E.CCLR,R4      ;CHECK IF "CLEAR CONTROLLER DID NOT
                                                                ;  CLEAR ERROR"
026572 001403                           BEQ     3$              ;NO, CHECK OTHER CONTROLLER ERRORS
026574 104401 061156                    TYPE    ,ERR001
026600 000473                           BR      10$             ;DISPLAY REGISTERS

026602 032704 000002            3$:     BIT     #E.NOAT,R4      ;CHECK IF "NO ATTENTION IN RKASOF"
026606 001403                           BEQ     4$              ;NO, CHECK THE OTHER CONTROLLER ERRORS
026610 104401 061223                    TYPE    ,ERR002
026614 000465                           BR      10$             ;DISPLAY REGISTERS

026616 032704 000010            4$:     BIT     #E.UDAT,R4      ;CHECK IF "UNEXPECTED DATA TYPE ERROR"
026622 001403                           BEQ     5$              ;NO, CHECK OTHER CONTROLLER ERRORS
026624 104401 061252                    TYPE    ,ERR003
026630 000457                           BR      10$             ;DISPLAY REGISTERS

026632 032704 000020            5$:     BIT     #E.CLAT,R4      ;CHECK IF "ATTENTION DID NOT RESET
                                                                ;  WITH DRIVE CLEAR"
026636 001403                           BEQ     6$              ;NO, CHECK OTHER CONTROLLER ERRORS
026640 104401 061305                    TYPE    ,ERR004
026644 000451                           BR      10$             ;DISPLAY REGISTERS

026646 032704 000100            6$:     BIT     #E.ILLD,R4      ;CHECK IF "ILLEGAL DRIVER COMMAND"
026652 001403                           BEQ     7$              ;NO, CHECK IF DATA LATE WHEN UNLOADING HEADER
026654 104401 061351                    TYPE    ,ERR005
026660 000443                           BR      10$             ;DISPLAY REGISTERS

026662 032704 000400            7$:     BIT     #E.DLT,R4       ;CHECK IF "DATA LATE WHEN UNLOADING HEADER"
026666 001403                           BEQ     8$              ;NO, CHECK IF "CONTROLLER ERROR WHILE GATERING
                                                                ;  STATUS"
026670 104401 064207                    TYPE    ,ERR200
026674 000435                           BR      10$             ;DISPLAY REGISTERS

026676 032704 001000            8$:     BIT     #E.CERR,R4      ;CHECK IF "CONTROLLER ERROR DURING
                                                                ;  DRIVE SERVICING"
026702 001425                           BEQ     9$              ;NO, CHECK IF CONTROLLER TIME OUT
026704 032737 000040 001426             BIT     #DTYE,T.ER      ;SEE IF DRIVE TYPE ERROR
026712 001003                           BNE     13$             ;BR IF YES
026714 104401 064247                    TYPE    ,ERR201
026720 000423                           BR      10$             ;DISPLAY REGISTERS
026722 132765 000001 000210     13$:    BITB    #BIT0,P.ASSN(R5)        ;SEE IF IN PROCESS OF ASSIGNING
026730 001003                           BNE     14$             ;BR IF YES
026732 104401 064247                    TYPE    ,ERR201         ;ELSE VALID ERROR
```

```
 6386  026736  000414                        BR      10$             ;DISPLAY REGS
 6387
 6388  026740  004737  027206        14$:    JSR     PC,FLPARM       ;FILL PARAM BLOCK WITH CORRECT RK06-07 DATA
 6389  026744  012737  000001  027334        MOV     #1,NEWFLG       ;SET FLAG TO SHOW DTYE ERROR ON ASSIGNING
 6390  026752  012604                        MOV     (SP)+,R4        ;RESTORE R4
 6391  026754  000427                        BR      12$
 6392
 6393  026756  032704  040000        9$:     BIT     #E.CMTO,R4      ;CHECK IF CONTROLLER TIME OUT
 6394  026762  001414                        BEQ     20$             ;NO, CHECK IF DRIVE DETECTED DRIVE BUS PARITY ERROR
 6395  026764  104401  064534                TYPE    ,ERR207
 6396
 6397  026770  104401  001165        10$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
 6398  026774  004737  044046                JSR     PC,PRITIM       ;PRINT TIME
 6399  027000  004737  026030                JSR     PC,PRIREG       ;PRINT CONTROLLER REGISTERS
 6400  027004  104401  061131                TYPE    ,ERR000         ;TYPE FATAL ERROR
 6401  027010  000000                        HALT                    ;HANG ON FATAL ERROR
 6402  027012  000776                        BR      .-2
 6403
 6404  027014  032704  002000        20$:    BIT     #E.DPAR,R4      ;CHECK IF 'DRIVE DETECTED DRIVE BUS PARITY ERROR''
 6405  027020  001001                        BNE     30$             ;YES, SET DRPAR IN ERROR REGISTER
 6406  027022  000000                        HALT
 6407
 6408  027024  012604                30$:    MOV     (SP)+,R4        ;RESTORE R4
 6409  027026  052765  000010  000034        BIS     #DRPAR,P.ER(R5) ;SET DRIVE DETECTED DRIVE BUS$OTT> PARITY
 6410  027034  010346                12$:    MOV     R3,-(SP)        ;STORE R3 ON THE STACK
 6411  027036  005037  001464                CLR     0.WAIT          ;CLEAR WAITING FOR COMMAND COMPLETION
 6412  027042  116503  000000                MOVB    P.DRVN(R5),R3   ;GET DRIVE NUMBER
 6413  027046  005062  000026                CLR     RKMR1(R2)       ;CLEAR MAINTENANCE REGISTER 1
 6414  027052  010362  000010                MOV     R3,RKCS2(R2)    ;LOAD DRIVE NUMBER
 6415  027056  052762  000040  000010        BIS     #SCLR,RKCS2(R2) ;DO SUBSYS CLR
 6416  027064  105762  000000        35$:    TSTB    RKCS1(R2)       ;WAIT FOR READY
 6417  027070  001775                        BEQ     35$
 6418  027072  004737  047414                JSR     PC,I.STOR       ;STORE REGISTERS
 6419  027076  032737  100000  001410        BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 6420  027104  001413                        BEQ     37$             ;NO, CHECK IF ATTENTION CLEARED
 6421
 6422                          ;             CHECK FOR CONTROLLER TYPE ERROR
 6423  027106  032737  137400  001412        BIT     #UFE!MDS!PGE!NEM!NED!UPE!DLT,T.CS2
 6424  027114  001004                        BNE     36$
 6425  027116  032737  130761  001426        BIT     #ILC!DTYE!FMTE!ECH!BSE!HVRC!DTE!OPI!DCK,T.ER
 6426  027124  001412                        BEQ     40$
 6427
 6428  027126  104401  064247        36$:    TYPE    ,ERR201         ;TYPE CONTROLLER ERROR DURING DRIVE SERVICING
 6429  027132  000716                        BR      10$             ;GO REPORT ERROR
 6430
 6431  027134  136337  001514  001425  37$:  BITB    INTMSK(R3),T.ASOF+1 ;CHECK IF ATTENTION CLEARED
 6432  027142  001403                        BEQ     40$             ;YES, REPORT ERROR
 6433  027144  104401  061305                TYPE    ,ERR004         ;TYPE 'DRIVE CLEAR DID NOT RESET ATTENTION''
 6434  027150  000707                        BR      10$             ;GO REPORT ERROR
 6435
 6436  027152  146337  001514  001512  40$:  BICB    INTMSK(R3),W.TIME ;CLEAR TIMING CURRENTLY ON THIS DRIVE
 6437  027160  006303                        ASL     R3              ;MULTIPLY BY 2
 6438  027162  005063  001544                CLR     W.DRV(R3)       ;CLEAR WATCH DOG TIME
 6439  027166  012603                        MOV     (SP)+,R3        ;RESTORE R3
 6440  027170  005737  027334                TST     NEWFLG          ;SEE IF DTYE ERROR ON ASSIGNING
 6441  027174  001402                        BEQ     41$             ;BR IF NO
```

```
 6442  027176  000137  024400                    JMP     NORMAL          ;ELSE GO NORMAL RET
 6443  027202  000137  025470          41$:      JMP     ABNORM          ;GO TO ABNORMAL RETURN
 6444
 6445  027206  132765  000004  000007  FLPARM:   BITB    #B.CDT,P.CS1H(R5)       ;SEE IF RK06 PARMETER USED LAST
 6446  027214  001407                            BEQ     1$              ;BR IF YES
 6447  027216  142765  000004  000007            BICB    #B.CDT,P.CS1H(R5)       ;ELSE TRY RK06 PARAM BECAUSE OF DTYE
 6448  027224  012765  000632  000070            MOV     #632,P.MXCL(R5)
 6449  027232  000207                            RTS     PC
 6450
 6451  027234  152765  000004  000007  1$:       BISB    #B.CDT,P.CS1H(R5)       ;TRY RK07 PARAM BECAUSE OF DTYE
 6452  027242  012765  001456  000070            MOV     #1456,P.MXCL(R5)
 6453  027250  000207                            RTS PC
 6454
 6455  027252  132765  000004  000007  GETNUM:   BITB    #B.CDT,P.CS1H(R5)       ;SEE IF RK06
 6456  027260  001412                            BEQ      1$             ;BR IF YES
 6457
 6458  027262  012737  001456  027332            MOV     #814.,LSTCYL    ;ELSE LOAD RK07 VALUES
 6459  027270  012737  151000  027336            MOV     #<814.*22.*3.>+<22.*2.>,HOLD
 6460  027276  012737  151011  027340            MOV     #<814.*22.*3.>+<22.*2.>+1,HOLD1
 6461  027304  000207                            RTS     PC
 6462
 6463  027306  012737  000632  027332  1$:       MOV     #410.,LSTCYL    ;LOAD RK06 VALUES
 6464  027314  012737  064740  027336            MOV     #<410.*22.*3.>+<22.*2.>,HOLD
 6465  027322  012737  064741  027340            MOV     #<410.*22.*3.>+<22.*2.>+1,HOLD1
 6466  027330  000207                            RTS     PC
 6467
 6468  027332  000000                  LSTCYL:   0
 6469  027334  000000                  NEWFLG:   0
 6470  027336  000000                  HOLD:     0
 6471  027340  000000                  HOLD1:    0
 6472
```

```
                                        .SBTTL  CONTROLLER ERROR DUE TO REIGSTER INCONSISTANCIES

027342  004737  052004          CNTRL1: JSR     PC,Q.RMOV       ;REMOVE PARAMETER BLOCK FROM THE
                                                                ;  COMMAND INITIATION  JEUE
027346  032777  002000  151564          BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
027354  001402                          BEQ     1$              ;NO, CONTINUE
027356  104401  001160                  TYPE    ,$BELL          ;RING BELL
027362  032777  020000  151550  1$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
027370  001404                          BEQ     2$              ;NO PRINT ERROR MESSAGE
027372  032777  040000  151540          BIT     #SW14,@SWR      ;CHECK IF CYCLE ON OPERATION
027400  001050                          BNE     10$             ;YES, CHECK IF LOOP ON OPERATION
027402  032737  000001  001564  2$:     BIT     #BIT0,ERCONT    ;CHECK IF "CONTROLLER ERROR NOT FLAGGED"
027410  001402                          BEQ     3$
027412  104401  061707                  TYPE    ,ERR013
027416  032737  000002  001564  3$:     BIT     #BIT1,ERCONT    ;CHECK IF 'WORD COUNT NOT 0"
027424  001402                          BEQ     4$
027426  104401  061746                  TYPE    ,ERR014
027432  032737  000004  001564  4$:     BIT     #BIT2,ERCONT    ;CHECK IF 'BUS ADDRESS INCORRECT"
027440  001402                          BEQ     5$
027442  104401  061777                  TYPE    ,ERR015
027446  032737  000010  001564  5$:     BIT     #BIT3,ERCONT    ;CHECK IF "CYLINDER,TRACK,SECTOR
                                                                ;  INCORRECT"
027454  001402                          BEQ     7$
027456  104401  062023                  TYPE    ,ERR016
027462  032777  020000  151450  7$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
027470  001014                          BNE     10$             ;YES, CHECK IF HALT ON ERROR
027472  004737  044046                  JSR     PC,PRITIM       ;PRINT TIME
027476  004737  026252                  JSR     PC,PRISTT       ;PRINT RK06 REGISTERS
027502  004737  030422                  JSR     PC,PRI002       ;PRINT CURRENT SUPPLIED COMMAND
027506  032737  000001  001564          BIT     #BIT0,ERCONT    ;CHECK IF CONTROLLER ERROR
                                                                ;  NOT FLAGGED
027514  001402                          BEQ     10$             ;NO, DO NOT PRINT PREVOUS COMMAND
027516  004737  030244                  JSR     PC,PRI001       ;PRINT PREVIOUS COMMAND
027522  032777  100000  151410  10$:    BIT     #SW15,@SWR      ;CHECK IF HALT ON ERROR
027530  001010                          BNE     11$             ;YES, HALT
027532  032777  040000  151400          BIT     #SW14,@SWR      ;CHECK IF LOOP OPERATION
027540  001005                          BNE     15$             ;YES, CLEAR ERROR
027542  104401  061131                  TYPE    ,ERR000         ;PRINT "FATAL ERROR"
027546  104401  061426                  TYPE    ,ERR007         ;PRINT "SET SWITCH 14 TO CYCLE ON ERROR"
027552  000000                  11$:    HALT
027554  012737  100000  001564  15$:    MOV     #BIT15,ERCONT   ;SET CONTROLLER ERROR PRESENT
027562  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
027570  005062  000026                  CLR     RKMR1(R2)       ;CLEAR MAINTANENCE REGISTER 1
027574  112737  000005  001410          MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND INTO TEMPORARY CS1
027602  004037  047152                  JSR     R0,I.ISSU       ;GO ISSUE DRIVE CLEAR
027606  027616                          20$                     ;ERROR RETURN
027610  004037  051564                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON COMMAND
027614  001324                          CINITQ                  ;  INITIATION QUEUE
027616  000207                  20$:    RTS     PC              ;RETURN
```

```
 6522                                   .SBTTL  PRINT WHOLE DATA BUFFER
 6523
 6524  027620  122765  000121  000123  PRIBUF: CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
 6525  027626  001017                          BNE     5$                 ;NO, RETURN
 6526  027630  032765  000010  000150          BIT     #BIT3,P.DSTT(R5) ;CHECK IF TIME TO PRINT BUFFER
```

```
 6527  027636  001013                        BNE     5$                ;NO, RETURN
 6528  027640  052765  000010  000150        BIS     #BIT3,P.DSTT(R5)  ;SET BUFFER ATTEMPTED TO BE PRINTED
 6529  027646  032777  020000  151264        BIT     #SW13,@SWR        ;CHECK IF INHIBIT PRINT OUT
 6530  027654  001004                        BNE     5$                ;YES, RETURN
 6531  027656  032777  000010  151254        BIT     #SW3,@SWR         ;CHECK IF PRINT SECTOR BEFORE BEGINNING
 6532                                                                  ;  ERROR RECOVERY
 6533  027664  001001                        BNE     10$               ;YES, PRINT SECTOR IN ERROR
 6534  027666  000207                5$:     RTS     PC                ;RETURN
 6535
 6536  027670  010146                10$:    MOV     R1,-(SP)          ;STORE R1 ON THE STACK
 6537  027672  010346                        MOV     R3,-(SP)          ;STORE R3 ON THE STACK
 6538  027674  010446                        MOV     R4,-(SP)          ;STORE R4 ON THE STACK
 6539  027676  104401  064717                TYPE    ,ERR213           ;TYPE ADDDRESS OF BEGINNING OF
 6540                                                                  ;  SECTOR IN ERROR
 6541  027702  013703  001634                MOV     RTYBA,R3          ;GET ADDRESS OF START OF SCTOR
 6542  027706  013704  001636                MOV     RTYWC,R4          ;GET WORD COUNT
 6543  027712  010346                        MOV     R3,-(SP)          ;STORE STARTING ADDRESS FOR PRINT OUT
 6544  027714  004737  053224                JSR     PC,BINOCT         ;CONVERT OF OCTAL
 6545  027720  104401  064770                TYPE    ,ERR214           ;TYPE DATA READ
 6546  027724  012701  000010        15$:    MOV     #8.,R1            ;SET 8 COLUMNS ON PAGE
 6547  027730  104401  001165                TYPE    ,$CRLF            ;TYPE <CR><LF>
 6548  027734  012346                20$:    MOV     (R3)+,-(SP)       ;GET NEXT WORD OF SECTOR
 6549  027736  004737  053224                JSR     PC,BINOCT         ;CONVERT TO OCTAL
 6550  027742  005304                        DEC     R4                ;CHECK IF FINISHED
 6551  027744  001405                        BEQ     30$               ;YES, RESTORE REGISTERS
 6552  027746  005301                        DEC     R1                ;CHECK IF END OF LINE
 6553  027750  001765                        BEQ     15$               ;YES, INITIALIZE COLUMN COUNT
 6554  027752  104401  061126                TYPE    ,BLANKS           ;TYPE 2 BLANKS
 6555  027756  000766                        BR      20$               ;GET NEXT WORD
 6556
 6557  027760  104401  001165        30$:    TYPE    ,$CRLF            ;TYPE <CR><LF>
 6558  027764  012604                        MOV     (SP)+,R4          ;RESTORE R4
 6559  027766  012603                        MOV     (SP)+,R3          ;RESTORE R3
 6560  027770  012601                        MOV     (SP)+,R1          ;RESTORE R1
 6561  027772  000207                        RTS     PC                ;RETURN
```

```
                                .SBTTL  PRINT DRIVE AND PACK SERIAL NUMBERS

030074  012701  057541          PRISER: MOV     R1,-(SP)        ;STORE R1 ON STACK
030076  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
030100  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
030102  012701  057176                  MOV     #STT100,R1      ;GET ADDRESS OF ASCII STRING
030106  016503  000224                  MOV     P.SERL(R5),R3   ;GET DRIVE SERIAL NUMBER
030112  006103                          ROL     R3              ;SHIFT DRIVE SERIAL NUMBER 4 BITS LEFT
030114  006103                          ROL     R3
030116  006103                          ROL     R3
030120  006103                          ROL     R3
030122  012704  000003                  MOV     #3,R4           ;LOAD NUMBER OF DIGITS OF SERIAL NUMBER
030126  105011                  2$:     CLRB    (R1)            ;CLEAR NEXT BYTE FOR PRINT OUT
030130  006103                          ROL     R3              ;SHIFT IN NEXT 4 BITS
030132  106111                          ROLB    (R1)
030134  006103                          ROL     R3
030136  106111                          ROLB    (R1)
030140  006103                          ROL     R3
030142  106111                          ROLB    (R1)
030144  006103                          ROL     R3
030146  106111                          ROLB    (R1)
030150  152721  000060                  BISB    #60,(R1)+       ;CHANGE BCD TO DECIMAL
030154  005304                          DEC     R4              ;CHECK IF FINISHED
030156  001363                          BNE     2$              ;CONVERT NEXT DIGIT
030160  104401  057153                  TYPE    ,STT000         ;TYPE 'DRIVE SERIAL NUMBER''
030164  016503  000226                  MOV     P.PKSR(R5),R3   ;GET LEAST SIGNIFICANT BITS OF
                                                                ;  PACK SERIAL NUMBER
030170  016504  000230                  MOV     P.PKSR+2(R5),R4 ;GET MOST SIGNIFICANT BITS OF
                                                                ;  PACK SERIAL NUMBER
030174  012701  057541                  MOV     #STT012,R1      ;GET ADDRESS OF ASCII STRING
030200  105011                          CLRB    (R1)            ;CLEAR FIRST BYTE OF PRINT OUT
030202  000404                          BR      7$              ;START CONVERSION TO OCTAL

030204  105011          5$:     CLRB    (R1)            ;CLEAR NEXT BYTE OF PRINT OUT
030206  006103                  ROL     R3              ;SHIFT IN NEXT THREE BITS
030210  006104                  ROL     R4
030212  106111                  ROLB    (R1)
030214  006103          7$:     ROL     R3
030216  006104                  ROL     R4
030220  106111                  ROLB    (R1)
030222  006103                  ROL     R3
030224  006104                  ROL     R4
030226  106111                  ROLB    (R1)
030230  152721  000060          BISB    #60,(R1)+       ;MAKE IT ASCII
030234  022701  057554          CMP     #STT013,R1      ;CHECK IF FINISHED
030240  101361                  BHI     5$              ;NO, CONTINUE CONVERSION
030242  104401  057514          TYPE    ,STT011         ;TYPE PACK SERIAL NUMBER
030246  012604                  MOV     (SP)+,R4        ;RESTORE R4
030250  012603                  MOV     (SP)+,R3        ;RESTORE R3
030252  012601                  MOV     (SP)+,R1        ;RESTORE R1
030254  000207                  RTS     PC              ;RETURN
```

```
 6613                                   .SBTTL  PRINT BEGINNING OF ERROR HEADER
 6614
 6615  030156  032777  002000  150754  PRI000: BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
 6616  030164  001402                          BEQ     1$              ;NO, CHECK IF INHIBIT PRINT OUT
 6617  030166  104401  001160                  TYPE    ,$BELL          ;RING BELL
 6618  030172  032777  020000  150740  1$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
 6619  030200  001020                          BNE     20$             ;YES, INCREMENT CONTROLLER ERROR
 6620  030202  116537  000000  056755          MOVB    P.DRVN(R5),OPR011 ;STORE DRIVE NUMBER
 6621  030210  152737  000060  056755          BISB    #60,OPR011      ;MAKE IT ASCII
 6622  030216  104401  056736                  TYPE    ,OPR010         ;TYPE "DRIVE NUMBER N"
 6623  030222  132765  000001  000210          BITB    #BIT0,P.ASSN(R5)        ;SEE IF SIZING FOR DRIVE
 6624  030230  001004                          BNE     20$             ;BR IF YES, TO RETURN
 6625  030232  004737  027774                  JSR     PC,PRISER       ;PRINT DRIVE AND PACK SERIAL NUMBERS
 6626  030236  004737  044046                  JSR     PC,PRITIM       ;PRINT TIME
 6627  030242  000207                  20$:     RTS     PC              ;RETURN
```

```
                                        .SBTTL  PRINT PREVIOUS COMMAND
                                PRI001: TYPE    ,ERR101         ;TYPE 'PREVIOUS CMND CYL TRK SEC  WRD CT'
000135                                  CMPB    #RDDATA,P.LCMD(R5) ;CHECK IF READ DATA
                                        BEQ     5$              ;YES, DO NOT PRINT PATTERN NUM
                                        TYPE    ,ERR102         ;PRINT 'PAT'
                                5$:     TYPE    ,$CRLF          ;PRINT <CR><LF>
                                        CLR     -(SP)           ;MAKE ROOM ON STACK
000135                                  MOVB    P.LCMD(R5),(SP) ;GET LAST COMMAND
                                        JSR     PC,BINOCT       ;CONVERT TO CCTAL
                                        TYPE    ,BLANKS         ;TYPE 2 BLANKS
000136                                  MOV     P.LCYL(R5),-(SP) ;GET LAST CYLINDER USED
                                        JSR     PC,BINOCT       ;CONVERT TO OCTAL
                                        TYPE    ,BLANKS         ;TYPE 2 BLANKS
                                        CLR     -(SP)           ;MAKE ROOM ON STACK
000141                                  MOVB    P.LTRK(R5),(SP) ;GET LAST TRACK USED
                                        JSR     PC,BINOCT       ;CONVERT TO OCTAL
                                        TYPE    ,BLANKS         ;TYPE 2 BLANKS
                                        CLR     -(SP)           ;MAKE ROOM ON STACK
000140                                  MOVB    P.LSEC(R5),(SP) ;GET LAST SECTOR USED
                                        JSR     PC,BINOCT       ;CONVERT TO OCTAL
                                        TYPE    ,BLANKS         ;TYPE 2 BLANKS
000142                                  MOV     P.LWC(R5),-(SP) ;STORE LAST WORD COUNT
                                        JSR     PC,BINOCT       ;CONVERT TO OCTAL
000135                                  CMPB    #RDDATA,P.LCMD(R5) ;CHECK IF PAT # VALID
                                        BEQ     10$             ;NO, DO NOT TYPE
                                        TYPE    ,BLANKS         ;TYPE 2 BLANKS
                                        CLR     -(SP)           ;MAKE ROOM ON STACK
000144                                  MOVB    P.LDPT(R5),(SP) ;GET LAST DATA PATTERN
                                        JSR     PC,BINOCT       ;CONVERT TO OCTAL
                                10$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
                                        RTS     PC              ;RETURN
```

```
                                        .SBTTL  PRINT CURRENT GENERATED COMMAND
                                PRI002: TYPE    ,ERR100         ;PRINT "CURRENT SUPPLIED COMMAND"
                                                                ;  CYL TRK SEC OFFSET
                                                                ;  BUS AD  WRD CT"
030426  122765  000121  000123        CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF PAT # IS TO BE PRINTED
030434  001402                        BEQ     5$              ;NO, DO NOT PRINT HEADER
030436  104401  064142                TYPE    ,ERR102         ;PRINT "PAT"
030442  104401  001165          5$:   TYPE    ,$CRLF          ;PRINT <CR><LF>
030446  005046                        CLR     -(SP)           ;MAKE ROOM ON STACK
030450  116516  000001                MOVB    P.CMND(R5),(SP) ;STORE COMMAND FOR PRINT OUT
030454  004737  053224                JSR     PC,BINOCT       ;CONVERT TO OCTAL
030460  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030464  016546  000002                MOV     P.CYLN(R5),-(SP) ;STORE CYLINDER
030470  004737  053224                JSR     PC,BINOCT       ;CONVERT TO OCTAL
030474  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030500  005046                        CLR     -(SP)           ;MAKE ROOM ON STACK
030502  116516  000005                MOVB    P.TRCK(R5),(SP) ;STORE TRACK
030506  004737  053224                JSR     PC,BINOCT       ;CONVERT TO OCTAL
030512  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030516  005046                        CLR     -(SP)           ;MAKE ROOM ON STACK
030520  116516  000004                MOVB    P.SECT(R5),(SP) ;STORE SECTOR
030524  004737  053224                JSR     PC,BINOCT       ;CONVERT TO OCTAL
030530  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030534  005046                        CLR     -(SP)           ;MAKE ROOM ON THE STACK
030536  122765  000021  000147        CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET
030544  101002                        BHI     10$             ;NO, INDICATE ZERO OFFSET
030546  116516  000006                MOVB    P.OFST(R5),(SP) ;STORE OFFSET FOR PRINT OUT
030552  004737  053224          10$:  JSR     PC,BINOCT       ;CONVERT TO OCTAL
030556  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030562  016546  000010                MOV     P.BALO(R5),-(SP) ;STORE BUS ADDRESS
030566  004737  053224                JSR     PC,BINOCT       ;COVERT TO OCTAL
030572  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030576  016546  000012                MOV     P.WC(R5),-(SP)  ;STORE WORD COUNT
030602  004737  053224                JSR     PC,BINOCT       ;COVERT TO OCTAL
030606  122765  000121  000123        CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF PAT # IS TO BE PRINTED
030614  001407                        BEQ     15$             ;NO, RETURN
030616  104401  061126                TYPE    ,BLANKS         ;TYPE 2 BLANKS
030622  005046                        CLR     -(SP)           ;MAKE ROOM ON THE STACK
030624  116516  000121                MOVB    P.DPAT(R5),(SP) ;STORE DATA PATTERN FOR PRINT OUT
030630  004737  053224                JSR     PC,BINOCT       ;CONVERT TO OCTAL
030634  104401  001165          15$:  TYPE    ,$CRLF          ;TYPE <CR><LF>
030640  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
                                ;*
                                ;*****************************************************************

030642  010446                  PRI100: MOV     R4,-(SP)        ;STORE R4 ON STACK
030644  012004                          MOV     (R0)+,R4        ;STORE ADDRESS OF PRINT CODE
030646  005204                          INC     R4              ;CALCULATE START OF MESSAGE
030650  010437  030662                  MOV     R4,2$           ;STORE ERROR MESSAGE ADDRESS
030654  004737  030156                  JSR     PC,PRI000       ;PRINT HEADER
030660  104401                          TYPE                    ;TYPE ERROR MESSAGE
030662  000000                  2$:     .WORD   0
030664  132765  000001  000210          BITB    #BIT0,P.ASSN(R5)        ;SEE IF SIZING FOR DRIVE
030672  001045                          BNE     12$             ;BR IF YES
030674  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
030700  032777  020000  150232          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINTOUT
030706  001033                          BNE     10$             ;YES, RETURN
030710  105765  000150                  TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
030714  100402                          BMI     3$              ;YES, GO PRINT CONTROLLER REGISTERS
030716  104401  064317                  TYPE    ,ERR202         ;TYPE ERROR DURING ERROR RECOVERY
030722  004737  026252          3$:     JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
030726  132744  000002                  BITB    #BIT1,-(R4)     ;CHECK IF PRINT DRIVE STATUS
030732  001407                          BEQ     5$              ;NO, CONTINUE
030734  132714  000040                  BITB    #BIT5,(R4)      ;CHECK IF QUESTIONABLE STATUS
030740  001402                          BEQ     4$              ;NO, DO NOT PRINT MESSAGE
030742  104401  064500                  TYPE    ,ERR206         ;PRINT "QUESTIONABLE DRIVE STATUS"
030746  004737  026076          4$:     JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
030752  132714  000100          5$:     BITB    #BIT6,(R4)      ;CHECK IF PRINT CURRENT GENERATED COMMAND
030756  001402                          BEQ     8$              ;NO, DO NOT PRINT CURRENT COMMAND
030760  004737  030422                  JSR     PC,PRI002       ;PRINT CURRENT GENERATED COMMAND
030764  132714  000004          8$:     BITB    #BIT2,(R4)      ;CHECK IF PRINT PREVIOUS COMMAND
030770  001402                          BEQ     10$             ;NO, CHECK IF HALT ON ERROR
030772  004737  030244                  JSR     PC,PRI001       ;PRINT PREVIOUS COMMAND
030776  004737  031426          10$:    JSR     PC,HLT000       ;TEST IF HALT ON ERROR
031002  005265  000206                  INC     P.NER(R5)       ;INCREMENT OTHER ERROR COUNT
031006  004737  023020          12$:    JSR     PC,BUFREL       ;RELEASE BUFFER
031012  004737  013162                  JSR     PC,DROP         ;DROP DRIVE FROM TEST SEQUENCE
031016  004737  022520                  JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
031022  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENTLY BEING PROCESSED
031026  012604                          MOV     (SP)+,R4        ;RESTORE R4
031030  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
                                ;*
                                ;;*****************************************************************

031032  010446                  PRI200: MOV     R4,-(SP)        ;STORE R4 ON STACK
031034  012004                          MOV     (R0)+,R4        ;GET PRINT CODE ERROR FLAGS
031036  132724  000001                  BITB    #BIT0,(R4)+     ;CHECK IF DATA TYPE ERROR
031042  001416                          BEQ     5$              ;NO, NON-DATA TYPE ERROR
031044  032765  000002  000150          BIT     #BIT1,P.DSTT(R5) ;CHECK IF DATA TYPE ERROR
                                                                ;  BEING PROCESSED
031052  001402                          BEQ     1$              ;NO, PRINT MESSAGE
031054  012604                          MOV     (SP)+,R4        ;RESTORE R4
031056  000200                          RTS     R0              ;RETURN

031060  052765  000002  000150  1$:     BIS     #BIT1,P.DSTT(R5) ;SET DATA TRANSFER ERROR OCCURRED
031066  100015                          BPL     10$             ;CHECK IF START OF ERROR RECOVERY
031070  052765  000001  000150          BIS     #BIT0,P.DSTT(R5) ;YES, SET SERVICING DATA ERROR
031076  000411                          BR      10$             ;GO REPORT ERROR

031100  032765  000004  000150  5$:     BIT     #BIT2,P.DSTT(R5) ;CHECK IF NON-DATA TYPE ERROR OCCURRED
031106  001402                          BEQ     6$              ;NO, REPORT ERROR
031110  012604                          MOV     (SP)+,R4        ;RESTORE R4
031112  000200                          RTS     R0              ;RETURN

031114  052765  000004  000150  6$:     BIS     #BIT2,P.DSTT(R5) ;SET NON-DATA ERROR OCCURRED
031122  004737  030156          10$:    JSR     PC,PRI000       ;TYPE ERROR HEADER
031126  032777  020000  150004          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
031134  001115                          BNE     30$             ;YES, INHIBIT PRINT OUT
031136  010437  031156                  MOV     R4,12$          ;STORE ERROR MESSAGE
031142  105765  000150                  TSTB    P.DSTT(R5)      ;CHECK IF ERROR DURING RECOVERY
031146  100402                          BMI     11$             ;NO, PRINT ERROR
031150  104401  064317                  TYPE    ,ERR202         ;PRINT "ERROR DURING ERROR RECOVERY"
031154  104401                  11$:    TYPE                    ;PRINT ERROR MESSAGE
031156  000000                  12$:    .WORD   0
031160  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
031164  004737  026252                  JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
031170  132744  000002                  BITB    #BIT1,-(R4)     ;CHECK IF PRINT DRIVE STATUS
031174  001407                          BEQ     22$             ;NO, CONTINUE
031176  132714  000040                  BITB    #BIT5,(R4)      ;CHECK IF QUESTIONABLE DRIVE STATUS
031202  001402                          BEQ     20$             ;NO, DO NOT PRINT MESSAGE
031204  104401  064500                  TYPE    ,ERR206         ;TYPE "QUESTIONABLE DRIVE STATUS"
031210  004737  026076          20$:    JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
```

```
6809  031214  132714  000100          22$:    BITB    #BIT6,(R4)          ;CHECK IF PRINT CURRENT COMMAND
6810  031220  001402                          BEQ     23$                 ;NO, CONTINUE
6811  031222  004737  030422                  JSR     PC,PRI002           ;PRINT CURRENTLY ISSUED COMMAND
6812  031226  132714  000004          23$:    BITB    #BIT2,(R4)          ;CHECK IF PRINT PREVIOUS COMMAND
6813  031232  001402                          BEQ     25$                 ;NO, CHECK IF PRINT HEADER
6814  031234  004737  030244                  JSR     PC,PRI001           ;PRINT PREVIOUS COMMAND
6815  031240  132714  000001          25$:    BITB    #BIT0,(R4)          ;CHECK IF DATA TYPE ERROR
6816  031244  001422                          BEQ     27$                 ;NO, DO NOT PRINT ECC
6817  031246  032765  100000  000034        BIT     #DCK,P.ER(R5)       ;CHECK IF DATA CHECK
6818  031254  001416                          BEQ     27$                 ;NO, DO NOT PRINT ECC
6819  031256  104401  063546                  TYPE    ,ERR067             ;TYPE ECC PAT, ECC POS
6820  031262  016546  000062                  MOV     P.EPAT(R5),-(SP)    ;GET PATTERN
6821  031266  004737  053224                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6822  031272  104401  061126                  TYPE    ,BLANKS
6823  031276  016546  000060                  MOV     P.EPOS(R5),-(SP)    ;GET POSITION
6824  031302  004737  053224                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6825  031306  104401  001165                  TYPE    ,$CRLF
6826  031312  132724  000020          27$:    BITB    #BIT4,(R4)+         ;CHECK IF PRINT HEADER
6827  031316  001424                          BEQ     30$                 ;NO, CHECK IF HALT ON ERROR
6828  031320  104401  064161                  TYPE    ,ERR111             ;TYPE HEADER DESCRIPTOR
6829  031324  013746  001646                  MOV     HEAD1,-(SP)         ;STORE FIRST WORD OF THE HEADER
6830  031330  004737  053224                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6831  031334  104401  061126                  TYPE    ,BLANKS             ;TYPE 2 BLANKS
6832  031340  013746  001650                  MOV     HEAD2,-(SP)         ;STORE SECOND WORD OF THE HEADER
6833  031344  004737  053224                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6834  031350  104401  061126                  TYPE    ,BLANKS             ;TYPE 2 BLANKS
6835  031354  013746  001652                  MOV     HEAD3,-(SP)         ;STORE THIRD WORD OF THE HEADER
6836  031360  004737  053224                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6837  031364  104401  001165                  TYPE    ,$CRLF              ;TYPE <CR><LF>
6838  031370  132744  000010          30$:    BITB    #BIT3,-(R4)         ;CHECK IF PRINT WHOLE SECTOR
6839  031374  001402                          BEQ     40$                 ;NO, CHECK IF HALT ON ERROR
6840  031376  004737  027620                  JSR     PC,PRIBUF           ;GO PRINT BUFFER
6841  031402  132714  000001          40$:    BITB    #BIT0,(R4)          ;CHECK IF DATA TYPE ERROR
6842  031406  001403                          BEQ     45$                 ;NO, CHECK IF HALT ON ERROR
6843  031410  042765  100200  000150        BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR
6844                                                                      ;  ENQUEUED
6845  031416  004737  031426          45$:    JSR     PC,HLT000           ;CHECK IF HALT ON ERROR
6846  031422  012604                          MOV     (SP)+,R4            ;RESTORE R4
6847  031424  000200                          RTS     R0                  ;RETURN
```

```
 6848                                    .SBTTL  HALT ON ERROR CHECK
 6849
 6850  031426  032777  100000  147504  HLT000: BIT     #SW15,@SWR      ;CHECK IF HALT ON ERROR
 6851  031434  001407                          BEQ     10$             ;NO, RETURN
 6852  031436  005737  001464                  TST     0.WAIT          ;CHECK IF WAITING FOR COMMAND
 6853                                                                  ;  COMPLETION
 6854  031442  001403                          BEQ     5$              ;NO, HALT
 6855  031444  105762  000000          1$:     TSTB    RKCS1(R2)       ;WAIT FOR READY
 6856  031450  001775                          BEQ     1$
 6857  031452  000000                  5$:     HALT
 6858  031454  000207                  10$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  OTHER ERROR STATISTICS GATHERING

031456  105765  000150          NERSTT: TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
031462  100057                          BPL     STT5$           ;NO, CONTINUE ERROR RECOVERY
031464  042765  100200  000150          BIC     #BIT15!BIT7,P.DSTT(R5) ;RESET FIRST ERROR AND ERROR ENQUEUED
031472  005265  000206                  INC     P.NER(R5)       ;INCREMENT OTHER ERRORS COUNT
031476  022765  000020  000206          CMP     #T.NER,P.NER(R5) ;CHECK IF ERROR THRESHOLD EXCEEDED
031504  103426                          BLO     STT2$           ;YES, CHECK IF DROP DRIVE
031506  000445                          BR      STT5$           ;NO, START ERROR RECOVERY SEQUENCE

                                        .SBTTL  SEEK INCOMPLETE STATISTICS GATHERING

031510  105765  000150          SKISTT: TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
031514  100042                          BPL     STT5$           ;NO, CONTINUE ERROR RECOVERY
031516  005265  000202                  INC     P.NSKI(R5)      ;INCREMENT NUMBER OF SEEK INCOMPLETES
031522  000405                          BR      STT1$           ;GO CHECK IF ERROR THRESHOLD EXCEEDED

                                        .SBTTL  OPERATION INCOMPLETE STATISTICS GATHERING

031524  105765  000150          OPISTT: TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
031530  100034                          BPL     STT5$           ;NO, CONTINUE ERROR PROCESSING
031532  005265  000204                  INC     P.NOPI(R5)      ;INCREMENT NUMBER OF OPERATION INCOMPLETES
031536  042765  100200  000150  STT1$:  BIC     #BIT15!BIT7,P.DSTT(R5) ;RESET FIRST ERROR AND ERROR ENQUEUED
031544  016546  000204                  MOV     P.NOPI(R5),-(SP) ;STORE NUMBER OF OPERATION INCOMPLETES
031550  066516  000202                  ADD     P.NSKI(R5),(SP) ;ADD SEEK INCOMPLETES
031554  026526  000104                  CMP     P.SKET(R5),(SP)+ ;CHECK IF ERROR THRESHOLD EXCEEDED
031560  103020                          BHIS    STT5$           ;NO, START ERROR RECOVERY
031562  032777  040020  147350  STT2$:  BIT     #SW4!SW14,@SWR  ;CHECK IF INHIBIT DRIVE DESELECTION
031570  001014                          BNE     STT5$           ;YES, START RECOVERY SEQUENCE
031572  104401  064445                  TYPE    ,ERR205         ;PRINT "ERROR THRESHOLD EXCEEDED"
031576  004737  023020                  JSR     PC,BUFREL       ;RELEASE BUFFER
031602  004737  013162                  JSR     PC,DROP         ;DROP DRIVE
031606  004737  022520                  JSR     PC,GETBUF       ;GET NEW COMMAND, DRIVE, AND BUFFER
031612  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENT BEING PROCESSED
031616  011000                          MOV     (R0),R0         ;INDICATE NO RECOVERY SEQUENCE IN PROGRESS
031620  000200                          RTS     R0              ;RETURN

031622  105765  000120          STT5$:  TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
031626  001010                          BNE     5$              ;YES, INDICATE UNRECOVERABLE RETURN
031630  105265  000146                  INCB    P.RECT(R5)      ;INCREMENT RETRY COUNT
031634  122765  000010  000146          CMPB    #8.,P.RECT(R5)  ;CHECK IF RETRY FINISHED
031642  103402                          BLO     5$              ;YES, CLEAN UP AFTER UNSUCCESSFUL RETRY
031644  005720                          TST     (R0)+           ;INDICATE RECOVERY SEQUENCE IN PROGRESS
031646  000200                          RTS     R0              ;RETURN

031650  004737  031660          5$:     JSR     PC,UNRECY       ;CLEAN UP FOR UNSUCCESSFUL RECOVERY
031654  011000                          MOV     (R0),R0         ;INDICATE NO RECOVERY SEQUENCE IN PROGRESS
031656  000200                          RTS     R0              ;RETURN
```

```
                                        .SBTTL  UNSUCESSFUL RECOVERY CLEANUP

031660  105037  001644          UNRECY: CLRB    RTYSCN          ;CLEAR RETRY SECTOR COUNT
031664  004737  023020                  JSR     PC,BUFREL       ;RELEASE BUFFER
031670  032765  000001  000150          BIT     #BIT0,P.DSTT(R5) ;CHECK IF DATA HARD ERROR
031676  001415                          BEQ     1$              ;NO, CHECK IF DRIVE IS BEING ASSIGNED
031700  005265  000200                  INC     P.HARD(R5)      ;INCREMENT HARD ERROR COUNT
031704  026565  000102  000200          CMP     P.UERT(R5),P.HARD(R5) ;CHECK IF HARD ERROR THRESHOLD EXCEEDED
031712  103007                          BHIS    1$              ;NO, CHECK IF DRIVE IS BEING ASSIGNED
031714  032777  040020  147216          BIT     #SW4!SW14,@SWR  ;INHIBIT DRIVE DESELECTION
031722  001003                          BNE     1$              ;YES, CHECK IF DRIVE IS BEING ASSIGNED
031724  104401  064445                  TYPE    ,ERR205         ;PRINT "ERROR THRESHOLD EXCEEDED
031730  000413                          BR      2$              ;GO DROP DRIVE

031732  105765  000210          1$:     TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
031736  001421                          BEQ     5$              ;NO, CLEAN UP ERROR
031740  132765  000010  000210          BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
031746  001404                          BEQ     2$              ;NO, DROP DRIVE
031750  132765  000200  000210          BITB    #BIT7,P.ASSN(R5) ;CHECK IF RECAL FOR READ OF
                                                                ;  PACK SERIAL NUMBER
031756  001441                          BEQ     10$             ;NO, ISSUE RECALIBRATE AND TRY AGAIN
031760  104401  064410          2$:     TYPE    ,ERR204         ;TYPE "ERROR RECOVERY UNSUCCESSFUL"
031764  004737  013162                  JSR     PC,DROP         ;DROP DRIVE FROM TEST SEQUENCE
031770  004737  022520                  JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
031774  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENT BEING PROCESSED
032000  000207                          RTS     PC              ;RETURN

032002  105765  000120          5$:     TSTB    P.IERC(R5)      ;CHECK IF INHIBIT RETRY
032006  001006                          BNE     6$              ;YES, DO NOT PRINT MESSAGE
032010  032777  020000  147122          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
032016  001002                          BNE     6$              ;YES, CONTINUE
032020  104401  064410                  TYPE    ,ERR204         ;TYPE "ERROR RECOVERY UNSUCCESSFUL"
032024  010446                  6$:     MOV     R4,-(SP)        ;STORE R4 ON STACK
032026  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER
032032  156437  001514  001513          BISB    INTMSK(R4),O.OVER ;INDICATE IMPLIED SEEKS ARE ALLOWED
032040  012604                          MOV     (SP)+,R4        ;RESTORE R4
032042  004037  051564                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN DRIVE
032046  001314                          AVAILQ                  ;  AVAILIABLE QUEUE
032050  004737  022520                  JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
032054  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENT BEING PROCESSED
032060  000207                          RTS     PC              ;RETURN

032062  122765  000010  000126  10$:   CMPB    #8.,P.RSEC(R5)  ;CHECK IF ALL 16-BIT MODE SECTORS HAVE BEEN TRIED
032070  101733                          BLOS    2$              ;YES, DROP DRIVE
032072  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
032074  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
032100  156437  001514  001513          BISB    INTMSK(R4),O.OVER ;INDICATE THAT IMPLIED SEEKS ARE ALLOWED
032106  012604                          MOV     (SP)+,R4        ;RESTORE R4
032110  152765  000200  000210          BISB    #BIT7,P.ASSN(R5) ;INDICATE THAT RECALIBRATE HAS BEEN ISSUED
032116  005065  000146                  CLR     P.RECT(R5)      ;CLEAR ERROR FLAGS
032122  005065  000150                  CLR     P.DSTT(R5)
032126  005065  000212                  CLR     P.ERR(R5)
032132  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE COMMAND
032140  004037  051564                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN COMMAND
032144  001324                          CINITQ                  ;  INITIATION QUEUE
032146  004737  022520                  JSR     PC,GETBUF       ;GO ALLOCATE MEMORY BUFFER
```

```
 6963  032152  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR PROCESSING IN PROGRESS
 6964  032156  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  ERROR RECOVERY SEQUENCE

032160  032765  000200  000212  ERCVY:  BIT     #BIT7,P.ERR(R5) ;CHECK IF UNSOLICITED INTERRUPT
032166  001441                          BEQ     3$              ;NO, CHECK IF ERROR INFORMATION STILL VALID
032170  032777  002000  146742          BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
032176  001402                          BEQ     1$              ;NO, CONTINUE PROCESSING
032200  104401  001160                  TYPE    ,$BELL          ;RING BELL
032204  032777  020000  146726  1$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
032212  001020                          BNE     2$              ;YES, CHECK IF HALT ON ERROR
032214  116537  000000  056755          MOVB    P.DRVN(R5),OPR011 ;LOAD DRIVE NUMBER FOR PRINT OUT
032222  152737  000060  056755          BISB    #60,OPR011      ;MAKE IT ASCII
032230  104401  056736                  TYPE    ,OPR010         ;TYPE "DRIVE N"
032234  004737  044046                  JSR     PC,PRITIM       ;PRINT TIME
032240  104401  062261                  TYPE    ,ERR026         ;TYPE "INT FORM DRV NOT UNDER TEST"
032244  004737  026252                  JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
032250  004737  026076                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
032254  004737  031426          2$:     JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
032260  005065  000212                  CLR     P.ERR(R5)       ;CLEAR ERROR FLAGS
032264  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
032270  000207                          RTS     PC              ;RETURN

032272  032765  000020  000212  3$:     BIT     #BIT4,P.ERR(R5) ;CHECK IF SERVICING DRIVE SEIZED TIME OUT
032300  001427                          BEQ     5$              ;NO, CONTINUE
032302  004737  030156                  JSR     PC,PRI000       ;PRINT HEADER
032306  104401  063351                  TYPE    ,ERR059         ;PRINT "TIME OUT BECAUSE DRIVE SEIZED"
032312  032777  020000  146620          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
032320  001002                          BNE     4$              ;YES, DO NOT PRINT CURRENT GENERATED COMMAND
032322  004737  030422                  JSR     PC,PRI002       ;PRINT CURRENT GENERATED COMMAND
032326  005265  000206          4$:     INC     P.NER(R5)       ;INCREMENT OTHER ERRORS
032332  004737  031426                  JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
032336  004737  023020                  JSR     PC,BUFREL       ;RELEASE BUFFER
032342  004737  013162                  JSR     PC,DROP         ;DROP DRIVE
032346  004737  022520                  JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
032352  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
032356  000207                          RTS     PC              ;RETURN

032360  032765  000400  000212  5$:     BIT     #BIT8,P.ERR(R5) ;CHECK IF TIME OUT BECAUSE HEADS DID NOT RELOAD
032366  001404                          BEQ     6$              ;NO, CONTINUE
032370  004037  030642                  JSR     R0,PRI100       ;PRINT TIME OUT BECAUSE HEADS DID NOT RELOAD
032374  063704                          ERR074
032376  000207                          RTS     PC              ;RETURN

032400  032765  000040  000212  6$:     BIT     #BIT5,P.ERR(R5) ;CHECK IF TIME OUT DURING DRIVE POSITIONING
032406  001425                          BEQ     7$               NO, CONTINUE
032410  042765  100000  000150          BIC     #BIT15,P.DSTT(R5) ;RESET ERROR ENQUEUED
032416  052765  000100  000150          BIS     #BIT6,P.DSTT(R5) ;SET READ STATUS ISSUED
032424  116537  000001  001666          MOVB    P.CMND(R5),POSCMD ;STORE COMMAND IN ERROR
032432  042765  000002  000014          BIC     #DRVPOS,P.PRST(R5) ;RESET DRIVE POSITIONING
032440  004737  037124                  JSR     PC,GETCUR       ;STORE PRESENT STATUS FOR PRINT OUT
032444  112765  000141  000001          MOVB    #RDSTAT,P.CMND(R5) ;GET DRIVE STATUS
032452  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE COMMAND IN COMMAND
032456  001324                          CINITQ                  ;  INITIATION QUEUE
032460  000207                          RTS     PC              ;RETURN

032462  032765  000100  000212  7$:     BIT     #BIT6,P.ERR(R5) ;CHECK IF ERROR WHILE ENQUEUED
032470  001404                          BEQ     8$              ;NO, CONTINUE
```

```
7021  032472  004037  030642                  JSR     R0,PRI100       ;PRINT HEADER
7022  032476  062141                          ERR022                  ;TYPE 'ERROR WHILE WAITING TO
7023                                                                  ;  REPORT ERROR''
7024  032500  000207                          RTS     PC              ;RETURN
7025
7026  032502  032765  004000  000016  8$:     BIT     #CTO,P.CS1(R5) ;CHECK IF CONTROLLER TIME OUT
7027  032510  001404                          BEQ     9$              ;NO, CONTINUE
7028  032512  004037  037500                  JSR     R0,CONERR       ;GO REPORT ERROR
7029  032516  062053                          ERR017
7030  032520  000207                          RTS     PC              ;RETURN
7031
7032  032522  032765  002000  000020  9$:     BIT     #PGE,P.CS2(R5)  ;CHECK IF PROGRAMMING ERROR
7033  032530  001404                          BEQ     10$             ;NO, CONTINUE
7034  032532  004037  037500                  JSR     R0,CONERR       ;PRINT HEADER
7035  032536  062107                          ERR020                  ;TYPE 'PROG ERROR''
7036  032540  000207                          RTS     PC              ;RETURN
7037
7038  032542  032765  000001  000034  10$:    BIT     #ILC,P.ER(R5)   ;CHECK IF ILLEGAL FUNCTION CODE
7039  032550  001404                          BEQ     11$             ;NO, CONTINUE
7040  032552  004037  037500                  JSR     R0,CONERR       ;PRINT HEADER
7041  032556  062122                          ERR021                  ;TYPE ''ILLEGAL FUNCTION CODE''
7042  032560  000207                          RTS     PC              ;RETURN
7043
7044  032562  032765  010000  000014  11$:    BIT     #DRVSZD,P.PRST(R5) ;CHECK FOR DRIVE AVAIL LOST
7045  032570  001404                          BEQ     12$             ;BR IF NOT
7046  032572  004037  030642                  JSR     R0,PRI100       ;PRINT HEADER
7047  032576  062400                          ERR029                  ;TYPE 'DRIVE BECAME NOT AVAIL''
7048  032600  000207                          RTS     PC              ;RETURN
7049
7050  032602  032765  010000  000020  12$:    BIT     #NED,P.CS2(R5)  ;CHECK FOR NON-EXISTENT DRIVE
7051  032610  001404                          BEQ     15$             ;NO, CONTINUE
7052  032612  004037  030642                  JSR     R0,PRI100       ;PRINT HEADER
7053  032616  062206                          ERR023                  ;TYPE 'NON-EXISTENT DRIVE''
7054  032620  000207                          RTS     PC              ;RETURN
7055
7056  032622  032765  000400  000020  15$:    BIT     #UFE,P.CS2(R5)  ;CHECK IF UNIT FIELD ERROR
7057  032630  001404                          BEQ     18$             ;NO, CONTINUE
7058  032632  004037  030642                  JSR     R0,PRI100       ;PRINT HEADER
7059  032636  062241                          ERR025                  ;TYPE 'UNIT FIELD ERROR''
7060  032640  000207                          RTS     PC              ;RETURN
7061
7062  032642  032765  020000  000016  18$:    BIT     #SPAR,P.CS1(R5) ;CHECK IF CONTROLLER DETECTED BAD PARITY
7063  032650  001004                          BNE     19$             ;YES, PRINT MESSAGE
7064  032652  032765  000010  000034          BIT     #DRPAR,P.ER(R5) ;CHECK IF DRIVE DETECTED BAD PARITY
7065  032660  001437                          BEQ     25$             ;NO, CONTINUE
7066  032662  004037  031032          19$:    JSR     R0,PRI200       ;PRINT HEADER
7067  032666  062225                          ERR024                  ;PRINT ''SERCON PARITY''
7068  032670  004037  031456                  JSR     R0,NERSTT       ;TAKE STATISTICS FOR OTHER ERRORS
7069  032674  033142                          60$                     ;ERROR RECOVERY SEQUENCE FINISHED RETURN
7070  032676  032765  020000  000016          BIT     #SPAR,P.CS1(R5) ;CHECK IF CONTROLLER DETECTD SERCON PARITY ERROR
7071  032704  001021                          BNE     22$             ;YES, GO RETRY COMMAND
7072  032706  032765  003400  000150          BIT     #BIT8!BIT9!BIT10,P.DSTT(R5) ;CHECK IF NO POSITIONING
7073  032714  001015                          BNE     22$             ;YES, RETRY COMMAND
7074  032716  132765  000007  000210          BITB    #BIT0!BIT1!BIT2,P.ASSN(R5) ;CHECK IF NO POSITION
7075                                                                  ;  DURING DRIVE ASSIGNMENT
7076  032724  001011                          BNE     22$             ;YES, RETRY COMMAND
```

```
7077  032726  042765  030000  000150          BIC     #BIT12!BIT13,P.DSTT(R5) ;RESET RE-SEEK AND RE-OFFSET
7078  032734  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE DRIVE
7079  032742  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE COMMAND
7080  032750  004037  051564          22$:    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN COMMAND
7081  032754  001324                          CINITQ                  ;  INITIATION QUEUE
7082  032756  000207                          RTS     PC              ;RETURN
7083
7084  032760  032765  004000  000020  25$:    BIT     #NEM,P.CS2(R5)  ;CHECK IF NON-EXISTENT MEMORY
7085  032766  001404                          BEQ     30$             ;NO,CONTINUE
7086  032770  004037  037500                  JSR     R0,CONERR       ;PRINT HEADER
7087  032774  062675                          ERR044                  ;PRINT 'NON-EXISTENT MEMORY''
7088  032776  000207                          RTS     PC              ;RETURN
7089
7090  033000  032765  020000  000020  30$:    BIT     #UPE,P.CS2(R5)  ;CHECK IF UNIBUS PARITY
7091  033006  001404                          BEQ     35$             ;NO, CONTINUE
7092  033010  004037  037500                  JSR     R0,CONERR       ;PRINT HEADER
7093  033014  062713                          ERR045                  ;PRINT 'UNIBUS PARITY''
7094  033016  000207                          RTS     PC              ;RETURN
7095
7096  033020  004037  037670          35$:    JSR     R0,TSTDRP       ;CHECK IF DROP DRIVE ERROR
7097  033024  033142                          60$                     ;YES, ERROR RETURN
7098  033026  032765  000002  000034          BIT     #SKI,P.ER(R5)   ;CHECK IF SEEK INCOMPLETE
7099  033034  001407                          BEQ     45$             ;NO, CONTINUE
7100  033036  004037  031032                  JSR     R0,PRI200       ;PRINT HEADER
7101  033042  062577                          ERR038                  ;PRINT ''SEEK INCOMPLETE''
7102  033044  004037  031510                  JSR     R0,SKISTT       ;TAKE SEEK INCOMPLETE STATISTICS
7103  033050  033142                          60$                     ;ERROR RECOVERY FINISHED RETURN
7104  033052  000412                          BR      48$             ;GO ISSUE RECALIBRATE
7105
7106  033054  032765  000040  000036  45$:    BIT     #DROT,P.DS(R5)  ;CHECK IF DRIVE OFF TRACK
7107  033062  001420                          BEQ     50$             ;NO, CONTINUE
7108  033064  004037  031032                  JSR     R0,PRI200       ;PRINT HEADER
7109  033070  062614                          ERR039                  ;PRINT 'DRIVE OFF TRACK''
7110  033072  004037  031456                  JSR     R0,NERSTT       ;TAKE STATISTICS ON OTHER ERRORS
7111  033076  033142                          60$                     ;END OF ERROR RECOVERY RETURN
7112  033100  042765  034000  000150  48$:    BIC     #BIT11!BIT12!BIT13,P.DSTT(R5) ;RESET STATUS
7113  033106  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
7114  033114  112765  000[illegible]  000001  MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE
7115  033122  000712                          BR      22$             ;ENQUEUE COMMAND
7116
7117  033124  032765  001000  000034  50$:    BIT     #COE,P.ER(R5)   ;CHECK IF CYLINDER ADDRESS OVERFLOW
7118  033132  001404                          BEQ     E0$             ;NO, CONTINUE
7119  033134  004037  037500                  JSR     R0,CONER[illegible]  ;PRINTER HEADER
7120  033140  063013                          ERR051                  ;PRINT ''CYLINDER ADDRESS OVERFLOW''
7121  033142  000207                  60$:    RTS     PC              ;RETURN
7122
7123  033144  005765  000022          E0$:    TST     P.WCR(R5)       ;CHECK IF FINAL WORD COUNT IS ZERO
7124  033150  001410                          BEQ     5$              ;YES, CONTINUE
7125  033152  026565  000022  000012          CMP     P.WCR(R5),P.WC(R5) ;CHECK THAT FINAL WORD COUNT IS LESS
7126                                                                  ;  THAN OR EQUAL TO INITIAL WORD COUNT
7127  033160  103004                          BHIS    5$              ;YES, CONTINUE
7128  033162  004037  037500                  JSR     R0,CONERR       ;TYPE 'WORD COUNT INVALID''
7129  033166  063567                          ERR070
7130  033170  000207                          RTS     PC              ;RETURN
7131
7132  033172  032765  001400  000016  5$:     BIT     #BA16!BA17,P.CS1(R5) ;CHECK IF BUS ADDRESS BITS 16 OR 17 SE1
```

```
7133  033200  001015                        BNE     8$              ;YES, REPORT ERROR
7134  033202  026565  000024  000010        CMP     P.BAR(R5),P.BALO(R5) ;CHECK THAT FINAL BUS ADDRESS IS GREATER
7135                                                                ;  THAN OR EQUAL TO INITIAL BUS ADDRESS
7136  033210  103411                        BLO     8$              ;NO, REPORT ERROR
7137  033212  016546  000012                MOV     P.WC(R5),-(SP)  ;PUT INITIAL WORD COUNT ON STACK
7138  033216  005416                        NEG     (SP)            ;MAKE IT POSITIVE
7139  033220  006316                        ASL     (SP)            ;MULTIPLY BY 2
7140  033222  066516  000010                ADD     P.BALO(R5),(SP) ;CALCULATE MAXIMUM FINAL ADDRESS
7141  033226  026526  000024                CMP     P.BAR(R5),(SP)+ ;MAKE SURE IT IS NOT TO LARGE
7142  033232  101404                        BLOS    10$             ;GOOD ADDRESS CONTINUE
7143  033234  004037  037500        8$:     JSR     R0,CONERR       ;TYPE 'BUS ADDRESS INVALID'
7144  033240  063612                        ERR071
7145  033242  000207                        RTS     PC              ;RETURN
7146
7147  033244  026565  000030  000002  10$:  CMP     P.DCYL(R5),P.CYLN(R5) ;CHECK THAT FINAL CYLINDER IS GREATER
7148                                                                ;  THAN OR EQUAL TO INITIAL CYLINDER
7149  033252  001405                        BEQ     15$             ;IF EQUAL CHECK TRACK/SECTOR
7150  033254  101014                        BHI     E1$             ;IF GREATER THAN, DETERMINE ERROR
7151  033256  004037  037500                JSR     R0,CONERR       ;TYPE 'CYLIN ADDRESS INVALID'
7152  033262  063636                        ERR072
7153  033264  000207                        RTS     PC              ;RETURN
7154
7155  033266  026565  000026  000004  15$:  CMP     P.DTS(R5),P.SECT(R5) ;CHECK TRACK/SECTOR GREATER THAN OR
7156                                                                ;  EQUAL TO INITIAL VALUES
7157  033274  103004                        BHIS    E1$             ;YES, GO INTO ERROR RECOVERY
7158  033276  004037  037500                JSR     R0,CONERR       ;TYPE 'SECTOR/TRACK INVALID'
7159  033302  063657                        ERR073
7160  033304  000207                        RTS     PC
7161
7162  033306  032765  010000  000034  E1$:  BIT     #DTE,P.ER(R5)   ;CHECK FOR DRIVE TIMING ERROR
7163  033314  001002                        BNE     2$              ;YES, PROCESS DRIVE TIMING ERROR
7164  033316  000137  034010                JMP     E2$             ;NO, CONTINUE
7165
7166  033322  105765  000147        2$:     TSTB    P.RERD(R5)      ;CHECK IF FIRST DATA ERROR IN
7167                                                                ;  RETRY SEQUENCE
7168  033326  001117                        BNE     30$             ;NO, REPORT ERROR
7169  033330  016537  000010  001634        MOV     P.BALO(R5),RTYBA ;STORE SUPPLIED BUS ADDRESS
7170  033336  016537  000012  001636        MOV     P.WC(R5),RTYWC ;STORE SUPPLIED WORD COUNT
7171  033344  005437  001636                NEG     RTYWC           ;MAKE IT POSITIVE
7172  033350  022737  000400  001636        CMP     #256.,RTYWC     ;CHECK IF LESS THAT OR EQUAL TO A SECTOR
7173  033356  103013                        BHIS    21$             ;YES, WORK WITH ONE SECTOR
7174  033360  026565  000026  000004        CMP     P.DTS(R5),P.SECT(R5) ;CHECK IF STARTING TRACK AND SECTOR
7175  033366  001021                        BNE     25$             ;NOT RETRY 2 SECTORS
7176  033370  026565  000030  000002        CMP     P.DCYL(R5),P.CYLN(R5) ;CHECK IF STARTING CYLINDER
7177  033376  001015                        BNE     25$             ;NO, RETRY 2 SECTORS
7178  033400  012737  000400  001636        MOV     #256.,RTYWC     ;SET WORD COUNT = 1 SECTOR
7179  033406  112737  000001  001644  21$:  MOVB    #1,RTYSCN       ;SET SECTOR COUNT = 1
7180  033414  016537  000004  001640        MOV     P.SECT(R5),RTYSEC ;LOAD SECTOR AND TRACK
7181  033422  016537  000002  001642        MOV     P.CYLN(R5),RTYCYL ;LOAD CYLINDER
7182  033430  000456                        BR      30$             ;GO REPORT ERROR
7183
7184  033432  016537  000026  001640  25$:  MOV     P.DTS(R5),RTYSEC ;SET SECTOR AND TRACK
7185  033440  016537  000030  001642        MOV     P.DCYL(R5),RTYCYL ;SET CYLINDER
7186  033446  105737  001640                TSTB    RTYSEC          ;CHECK IF SECTOR = 0
7187  033452  001403                        BEQ     26$             ;YES, GO DECREMENT TRACK
7188  033454  105337  001640                DECB    RTYSEC          ;DECREMENT SECTOR
```

```
7189  033460  000416                          BR      28$             ;DETERMINE WORD COUNT
7190
7191  033462  112737  000025  001640  26$:    MOVB    #21.,RTYSEC     ;LOAD SECTOR = 21
7192  033470  105737  001641                  TSTB    RTYTRK          ;CHECK IF TRACK 0
7193  033474  001403                          BEQ     27$             ;YES, GO DECREMENT CYLINDER
7194  033476  105337  001641                  DECB    RTYTRK          ;DECREMENT TRACK
7195  033502  000405                          BR      28$             ;DETERMINE WORD COUNT
7196
7197  033504  112737  000002  001641  27$:    MOVB    #2,RTYTRK       ;LOAD TRACK = 2
7198  033512  005337  001642                  DEC     RTYCYL          ;DECREMENT CYLINDER
7199  033516  004737  037366          28$:    JSR     PC,CALWC        ;CALCULATE RETRY WORD COUNT
7200  033522  112737  000002  001644          MOVB    #2,RTYSCN       ;LOAD SECTOR COUNT = 2
7201  033530  022737  001000  001636          CMP     #512.,RTYWC     ;CHECK IF LESS THAN OR EQUAL
7202  033536  101004                          BHI     29$             ;   TWO SECTORS
7203  033540  012737  001000  001636          MOV     #512.,RTYWC     ;MAKE IT 512
7204  033546  000407                          BR      30$             ;GO REPORT ERROR
7205
7206  033550  022737  000400  001636  29$:    CMP     #256.,RTYWC     ;CHECK IF ONLY ONE SECTOR
7207  033556  103403                          BLO     30$             ;NO, GO REPORT ERROR
7208  033560  112737  000001  001644          MOVB    #1,RTYSCN       ;LOAD SECTOR COUNT = 1
7209
7210  033566  004037  031032          30$:    JSR     R0,PRI200       ;PRINT HEADER
7211  033572  062726                          ERR046                  ;PRINT DRIVE TIMING ERROR
7212  033574  105765  000120                  TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
7213  033600  001016                          BNE     35$             ;YES, GO CLEAN UP FOR NEXT COMMAND
7214  033602  105265  000147                  INCB    P.RERD(R5)      ;INCREMENT REREAD COUNT
7215  033606  122765  000043  000147          CMPB    #43,P.RERD(R5)  ;CHECK IF RETRY UNSUCCESSFUL
7216  033614  101410                          BLOS    35$             ;YES, GO INDICATE UNSUCCESSFUL
7217                                                                  ;   RECOVERY
7218  033616  122765  000121  000123          CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
7219  033624  001407                          BEQ     37$             ;YES, GO TO RECOVERY SEQUENCE
7220  033626  122765  000021  000147          CMPB    #21,P.RERD(R5)  ;CHECK IF RETRY UNSUCCESSFUL
7221  033634  101026                          BHI     40$             ;NO, GO TO RETRY SEQUENCE
7222  033636  004737  031660          35$:    JSR     PC,UNRECY       ;CLEAN UP FOR UNSUCCESSFUL RECOVERY
7223  033642  000207                          RTS     PC              ;RETURN
7224
7225  033644  122765  000021  000147  37$:    CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET +400 MICRO-INCHES
7226  033652  001017                          BNE     40$             ;NO, CONTINUE RECOVERY
7227  033654  122737  000001  001644          CMPB    #1,RTYSCN       ;CHECK IF NUMBER OF SECTORS = 1
7228  033662  001404                          BEQ     38$             ;YES, DO OFFSET
7229  033664  122737  000025  001640          CMPB    #21.,RTYSEC     ;CHECK IF MID-TRANSFER SEEK
7230                                                                  ;   IS REQUIRED
7231  033672  001761                          BEQ     35$             ;YES, INDICATE UNSUCCESSFUL RETRY
7232  033674  010446                  38$:    MOV     R4,-(SP)        ;STORE R4 ON THE STACK
7233  033676  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
7234  033702  146437  001514  001513          BICB    INTMSK(R4),O.OVER ;DO NOT SEEK BEFORE DATA
7235                                                                  ;   TRANSFER (RESETTING OFFSET)
7236  033710  012604                          MOV     (SP)+,R4        ;RESTORE R4
7237
7238  033712  004037  037204          40$:    JSR     R0,DETOFF       ;DETERMINE IF OFFSET INSTRUCTION
7239  033716  034006                          50$                     ;OFFSET ISSUED RETURN
7240  033720  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;LOAD COMMAND
7241  033726  013765  001640  000004          MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK
7242  033734  013765  001642  000002          MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER
7243  033742  013765  001634  000010          MOV     RTYBA,P.BALO(R5) ;LOAD BUS ADDRESS
7244  033750  013765  001636  000012          MOV     RTYWC,P.WC(R5)  ;LOAD WORD COUNT
```

```
7245  033756  005465  000012                  NEG     P.WC(R5)            ;MAKE IT NEGATIVE
7246  033762  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
7247  033770  001003                          BNE     45$                 ;NO, ISSUE COMMAND
7248  033772  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;YES, SET ISSUE WRITE COMMAND
7249  034000  004037  051564          45$:    JSR     R0,Q.PUSH           ;ENQUEUE PARAMETER BLOCK IN
7250  034004  001324                          CINITQ                      ;  COMMAND INITIATION QUEUE
7251  034006  000207          50$:    RTS     PC                  ;RETURN
7252
7253  034010  032765  020400  000034  E2$:    BIT     #OPI!HVRC,P.ER(R5) ;CHECK IF HEADER TYPE ERROR
7254  034016  001002                          BNE     2$                  ;YES, PROCESS ERROR
7255  034020  000137  034346                  JMP     E3$                 ;NO, CONTINUE
7256
7257  034024  004737  037124          2$:     JSR     PC,GETCUR           ;GET CURRENT REGISTER STATUS
7258  034030  112737  000001  001644          MOVB    #1,RTYSCN           ;LOAD RETRY SECTOR COUNT TO 1
7259  034036  016537  000026  001640          MOV     P.DTS(R5),RTYSEC ;STORE CURRENT SECTOR AND TRACK
7260  034044  016537  000030  001642          MOV     P.DCYL(R5),RTYCYL ;STORE CURRENT SECTOR
7261  034052  016537  000010  001634          MOV     P.BALO(R5),RTYBA ;GET INITIAL BUS ADDRESS
7262  034060  016537  000012  001636          MOV     P.WC(R5),RTYWC  ;GET INITIAL WORD COUNT
7263  034066  005437  001636                  NEG     RTYWC               ;MAKE IT POSITIVE
7264  034072  022737  000400  001636          CMP     #256.,RTYWC     ;CHECK IF LESS THEN OR EQUAL TO ONE SECTOR
7265  034100  103021                          BHIS    20$                 ;RETRY COMMAND
7266  034102  026565  000026  000004          CMP     P.DTS(R5),P.SECT(R5) ;CHECK IF CURRENT SECTOR AND TRACK EQUALS
7267                                                                      ;  INITIAL TRACK AND SECTOR
7268  034110  001004                          BNE     15$                 ;NO, CALCULATE DESIRED SECTOR
7269  034112  026565  000030  000002          CMP     P.DCYL(R5),P.CYLN(R5) ;CHECK IF CURRENT CYLINDER EQUALS
7270                                                                      ;  INITIAL CYLINDER
7271  034120  001406                          BEQ     18$                 ;YES, USE INITIAL CYLINDER, TRACK, AND SECTOR
7272  034122  004737  037366          15$:    JSR     PC,CALWC            ;CALCULATE WORD COUNT
7273  034126  022737  000400  001636          CMP     #256.,RTYWC     ;CHECK IF WORD COUNT LESS THAN OR EQUAL
7274                                                                      ;  TO ONE SECTOR
7275  034134  103003                          BHIS    20$                 ;YES, USE CALCULATED WORD COUNT
7276  034136  012737  000400  001636  18$:     MOV     #256.,RTYWC     ;USE 256 WORDS
7277  034144  016546  000052          20$:    MOV     P.B10(R5),-(SP) ;STORE DRIVE CYLINDER ADDRESS
7278  034150  042716  160017                  BIC     #^C<M.CADD>,(SP) ;THROW AWAY PARITY AND MESSAGE TYPE
7279  034154  006216                          ASR     (SP)                ;SHIFT 4 BITS RIGHT
7280  034156  006216                          ASR     (SP)
7281  034160  006216                          ASR     (SP)
7282  034162  006216                          ASR     (SP)
7283  034164  022637  001642                  CMP     (SP)+,RTYCYL    ;CHECK IF CYLINDER IS CORRECT
7284  034170  001420                          BEQ     30$                 ;YES, ISSUE READ HEADER
7285  034172  004037  031032                  JSR     R0,PRI200           ;REPORT ERROR
7286  034176  062631                          ERR040                      ;TYPE 'MISPOS''
7287  034200  004037  031524                  JSR     R0,OPISTT           ;GET OPI/MISPOSITIONING STATISTICS
7288  034204  034344                          60$                         ;ERROR THRESHOLD EXCEEDED RETURN
7289  034206  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
7290  034214  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE IN PARAMETER BLOCK
7291  034222  004037  051564                  JSR     R0,Q.PUSH           ;PUT PARAMETER BLOCK IN COMMAND
7292  034226  001324                          CINITQ                      ;  INITIATION QUEUE
7293  034230  000207                          RTS     PC                  ;RETURN
7294
7295  034232  010446          30$:    MOV     R4,-(SP)            ;STORE R4 ON THE STACK
7296  034234  113704  001641                  MOVB    RTYTRK,R4           ;GET TRACK FOR INDEX
7297  034240  136465  001507  000057          BITB    H.HEAD(R4),P.B11+1(R5) ;CHECK IF CORRECT HEAD IS SELECTED
7298  034246  001013                          BNE     35$                 ;YES, CONTINUE
7299  034250  012604                          MOV     (SP)+,R4            ;RESTORE R4
7300  034252  004037  031032                  JSR     R0,PRI200           ;PRINT HEAD SELECT ERROR
```

```
 7301  034256  063503                        ERR065
 7302  034260  004037  031524                JSR     R0,OPISTT       ;RECORD AS AN OPI
 7303  034264  034344                        60$                     ;ERROR THRESHOLD EXCEEDED RETURN
 7304  034266  004037  051564                JSR     R0,Q.PUSH       ;REISSUE COMMAND
 7305  034272  001324                        CINITQ
 7306  034274  000207                        RTS     PC              ;RETURN
 7307
 7308  034276  012604                35$:    MOV     (SP)+,R4        ;RESTORE R4
 7309  034300  052765  000020  000150        BIS     #BIT4,P.DSTT(R5) ;SET READ ALL HEADERS ISSUED
 7310  034306  013765  001640  000004        MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK
 7311  034314  013765  001642  000002        MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER
 7312  034322  012765  066404  000010        MOV     #HDBUFF,P.BALO(R5) ;LOAD ADDRESS OF HEADER BUFFER
 7313  034330  112765  000164  000001        MOVB    #RDALHD,P.CMND(R5) ;LOAD COMMAND
 7314  034336  004037  051564                JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN COMMAND
 7315  034342  001324                        CINITQ                  ;  INITIATION QUEUE
 7316  034344  000207                60$:    RTS     PC              ;RETURN
 7317
 7318  034346  032765  000200  000034  E3$:  BIT     #BSE,P.ER(R5)   ;CHECK FOR BAD SECTOR ERROR
 7319  034354  001403                        BEQ     5$              ;NO, CONTINUE
 7320  034356  004737  042534                JSR     PC,BDSECT       ;SERVICE BAD SECTOR
 7321  034362  000207                        RTS     PC              ;RETURN
 7322
 7323  034364  032765  100000  000020  5$:   BIT     #DLT,P.CS2(R5)  ;CHECK IF DATA LATE
 7324  034372  001444                        BEQ     E4$             ;NO, CONTINUE
 7325  034374  004737  030156                JSR     PC,PRI000       ;TYPE HEADER
 7326  034400  032777  020000  144532        BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
 7327  034406  001004                        BNE     10$             ;YES, CHECK IF HALT ON ERROR
 7328  034410  104401  062746                TYPE    ,ERR047         ;TYPE 'DATA LATE''
 7329  034414  004737  026252                JSR     PC,PRISTT       ;PRINT CONTROLLER STATUS
 7330  034420  004737  031426          10$:  JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
 7331  034424  005237  001626                INC     DLTCNT          ;INCREMENT DATA LATE COUNT
 7332  034430  022737  001000  001626        CMP     #T.DLT,DLTCNT   ;CHECK IF THRESHOLD EXCEEDED
 7333  034436  103006                        BHIS    15$             ;NO, REISSUE COMMAND
 7334  034440  104401  061131                TYPE    ,ERR000         ;TYPE FATAL ERROR
 7335  034444  104401  061534                TYPE    ,ERR009         ;TYPE DATA LATE THRESHOLD EXCEEDED
 7336  034450  000000                        HALT                    ;HANG
 7337  034452  000776                        BR      .-2
 7338
 7339  034454  105765  000150          15$:  TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
 7340  034460  100005                        BPL     20$             ;NO, REISSUE COMMAND
 7341  034462  042765  100200  000150        BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR ERROR ENQUEUED AND FIRST ERROR
 7342  034470  005037  001566                CLR     ERRPRO          ;CLEAR ERROR RECOVERY IN PROGRESS
 7343  034474  004037  051564          20$:  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7344  034500  001324                        CINITQ                  ;  INITIATION QUEUE
 7345  034502  000207                        RTS     PC              ;RETURN
 7346
 7347  034504  032765  100000  000034  E4$:  BIT     #DCK,P.ER(R5)   ;CHECK IF DATA CHECK
 7348  034512  001002                        BNE     2$              ;YES, START RECOVERY
 7349  034514  000137  035314                JMP     E5$             ;NO, CONTINUE
 7350
 7351  034520  112737  000001  001644  2$:   MOVB    #1,RTYSCN       ;SET SECTOR COUNT = 1
 7352  034526  016537  000004  001640        MOV     P.SECT(R5),R YSEC ;STORE RETRY SECOTOR AND TRACK
 7353  034534  016537  000002  001642        MOV     P.CYLN(R5),RTYCYL ;STORE RETRY CYLINDER
 7354  034542  016537  000010  001634        MOV     P.BALO(R5),RTYBA ;STORE RETRY BUS ADDRESS
 7355  034550  016537  000012  001636        MOV     P.WC(R5),RTYWC ;STORE RETRY WORD COUNT
 7356  034556  005437  001636                NEG     RTYWC           ;MAKE IT POSITIVE
```

```
 7357  034562  022737  000400  001636          CMP     #256.,RTYWC      ;CHECK IF LESS THAN OR EQUAL TO A SECTOR
 7358  034570  103043                          BHIS    15$              ;YES, CHECK IF ECC CORRECTION
 7359  034572  016537  000026  001640          MOV     P.DTS(R5),RTYSEC ;STORE PRESENT TRACK AND SECTOR
 7360  034600  016537  000030  001642          MOV     P.DCYL(R5),RTYCYL ;STORE PRESENT CYLINDER
 7361  034606  105737  001640                  TSTB    RTYSEC           ;CHECK IF SECTOR = 0
 7362  034612  001403                          BEQ     11$              ;YES, DECREMENT TRACK
 7363  034614  105337  001640                  DECB    RTYSEC           ;DECREMENT SECTOR
 7364  034620  000416                          BR      13$              ;GO CALCULATE WORD COUNT
 7365
 7366  034622  112737  000025  001640  11$:    MOVB    #21.,RTYSEC      ;SET SECTOR = 21
 7367  034630  105737  001641                  TSTB    RTYTRK           ;CHECK IF TRACK = 0
 7368  034634  001403                          BEQ     12$              ;YES, DECREMENT CYLINDER
 7369  034636  105337  001641                  DECB    RTYTRK           ;DECREMENT TRACK
 7370  034642  000405                          BR      13$              ;GO CALCULATE WORD COUNT
 7371
 7372  034644  112737  000002  001641  12$:    MOVB    #2,RTYTRK        ;SET TRACK = 2
 7373  034652  005337  001642                  DEC     RTYCYL           ;DECREMENT CYLINDER
 7374  034656  004737  037366          13$:    JSR     PC,CALWC         ;CALCULATE WORD COUNT
 7375  034662  022737  000400  001636          CMP     #256.,RTYWC      ;CHECK IF LESS THAN OR EQUAL TO ONE SECTOR
 7376  034670  103003                          BHIS    15$              ;YES, PERFORM ECC CORRECTION
 7377  034672  012737  000400  001636          MOV     #256.,RTYWC      ;MAKE IT ONE SECTOR
 7378  034700  004037  031032          15$:    JSR     R0,PRI200        ;PRINT HEADER
 7379  034704  062762                          ERR048                   ;PRINT DATA CHECK
 7380  034706  105765  000120                  TSTB    P.IERC(R5)       ;CHECK IF INHIBIT ERRROR RECOVERY
 7381  034712  001123                          BNE     40$              ;YES, INDICATE UNSUCESSFUL RECOVERY
 7382  034714  122765  000121  000123          CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
 7383  034722  001101                          BNE     30$              ;NO, DO NOT DO ECC CORRECTION
 7384  034724  032765  000100  000034          BIT     #ECH,P.ER(R5)    ;CHECK IF CORRECTABLE
 7385  034732  001075                          BNE     30$              ;NO, DO NOT DO ECC CORRECTION
 7386  034734  022765  010040  000060          CMP     #10040,P.EPOS(R5) ;CHECK IF ECC POSITION TOO LARGE
 7387  034742  101406                          BLOS    16$              ;YES, REPORT ERROR
 7388  034744  005765  000060                  TST     P.EPOS(R5)       ;CHECK IF ECC POSITION = 0
 7389  034750  001403                          BEQ     16$              ;YES, REPORT ERROR
 7390  034752  005765  000062                  TST     P.EPAT(R5)       ;CHECK IF ECC PATTERN = 0
 7391  034756  001004                          BNE     17$              ;NO, DO ECC CORRECTION
 7392  034760  004037  037500          16$:    JSR     R0,CONERR        ;TYPE "ECC LOGIC ERROR"
 7393  034764  063526                          ERR066
 7394  034766  000207                          RTS     PC               ;RETURN
 7395
 7396  034770  004737  026336          17$:    JSR     PC,ECCCOR        ;DO ECC CORRECTION
 7397  034774  105765  000210                  TSTB    P.ASSN(R5)       ;CHECK IF IN ASSIGNMENT SEQUENCE
 7398  035000  001017                          BNE     18$              ;YES, DO NOT CHECK ECC CORRECTION
 7399  035002  013765  001634  000214          MOV     RTYBA,P.ERHD(R5) ;LOAD START OF SECTOR FOR
 7400                                                                   ;  DATA COMPARISON
 7401  035010  013765  001636  000222          MOV     RTYWC,P.BFCP(R5) ;LOAD NUMBER OF WORDS OF
 7402  035016  013765  001636  000220          MOV     RTYWC,P.CMLG(R5) ;  SECTOR PRESENT
 7403  035024  112765  177777  000117          MOVB    #-1,P.ECMP(R5)   ;SET ECC COMPARE FLAG
 7404  035032  004737  023540                  JSR     PC,CMPBUF        ;CHECK ECC CORRECTION
 7405  035036  035046                          20$                      ;INDICATE ECC MISCORRECTION
 7406  035040  004737  036454          18$:    JSR     PC,SURECY        ;INDICATE SUCCESSFUL RECOVERY
 7407  035044  000207                          RTS     PC               ;RETURN
 7408
 7409  035046  032777  020000  144064  20$:    BIT     #SW13,@SWR       ;CHECK IF INHIBIT ERROR PRINT OUT
 7410  035054  001024                          BNE     30$              ;YES, GO TO RECOVERY SEQUENCE
 7411  035056  104401  063422                  TYPE    ,ERR060          ;TYPE "ECC MISCORRECTION"
 7412  035062  104401  064701                  TYPE    ,ERR211          ;TYPE "RKPOS"
```

```
7413  035066  016546  000060                    MOV     P.EPOS(R5),-(SP) ;STORE RKPOS FOR PRINT OUT
7414  035072  004737  053224                    JSR     PC,BINOCT       ;CONVERT TO OCTAL
7415  035076  104401  001165                    TYPE    ,$CRLF          ;TYPE <CR><LF>
7416  035102  104401  064710                    TYPE    ,ERR212         ;TYPE 'RKPAT'
7417  035106  016546  000062                    MOV     P.EPAT(R5),-(SP) ;STORE RKPAT FOR PRINT OUT
7418  035112  004737  053224                    JSR     PC,BINOCT       ;COVERT TO OCTAL
7419  035116  104401  001165                    TYPE    ,$CRLF          ;TYPE <CR><LF>
7420  035122  004737  036202                    JSR     PC,PRERWD       ;PRINT WORD IN ERROR
7421  035126  105265  000147          30$:      INCB    P.RERD(R5)      ;INCREMENT RETRY COUNT
7422  035132  122765  000043  000147            CMPB    #43,P.RERD(R5)  ;CHECK IF UNSUCCESSFUL RETRY
7423  035140  101410                            BLOS    40$             ;YES, INDICATE UNSUCESSFUL RETRY
7424  035142  122765  000121  000123            CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
7425  035150  001407                            BEQ     45$             ;YES, CHECK FOR OFFSET
7426  035152  122765  000021  000147            CMPB    #21,P.RERD(R5)  ;CHECK IF ABOUT TO ENTER OFFSET OPERATATIONS
7427  035160  101021                            BHI     50$             ;NO, DO REREAD
7428  035162  004737  031660          40$:      JSR     PC,UNRECV       ;INDICATE UNSUCCESSFUL RECOVERY
7429  035166  000207                            RTS     PC              ;RETURN
7430
7431  035170  122765  000021  000147  45$:      CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET +400 MICRO-INCHES
7432  035176  001007                            BNE     48$             ;NO, GO DETERMINE IF OFFSET IS TO
7433                                                                    ;  BE ISSUED
7434  035200  010446                            MOV     R4,-(SP)        ;STORE R4 ON STACK
7435  035202  116504  000000                    MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER AS INDEX
7436  035206  146437  001514  001513            BICB    INTMSK(R4),O.OVER ;RESET ISSUE IMPLIED SEEK
7437  035214  012604                            MOV     (SP)+,R4        ;RESTORE R4
7438  035216  004037  037204          48$:      JSR     R0,DETOFF       ;DETERMINE IF OFFSET IS TO BE ISSUED
7439  035222  035312                            60$                     ;OFFSET ISSUED RETURN
7440  035224  116565  000123  000001  50$:      MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE LAST COMMAND
7441  035232  013765  001640  000004            MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK
7442  035240  013765  001642  000002            MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER
7443  035246  013765  001634  000010            MOV     RTYBA,P.BALO(R5) ;LOAD BUS ADDRESS
7444  035254  013765  001636  000012            MOV     RTYWC,P.WC(R5)  ;LOAD WORD COUNT
7445  035262  005465  000012                    NEG     P.WC(R5)        ;MAKE IT NEGATIVE
7446  035266  122765  000131  000001            CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
7447  035274  001003                            BNE     55$             ;NO, GO ISSUE COMMAND
7448  035276  052765  000200  000014            BIS     #W.WCK,P.PRST(R5) ;ISSUE WRITE BEFORE WRITE CHECK
7449  035304  004037  051564          55$:      JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON COMMAND
7450  035310  001324                            CINITQ                  ;COMMAND INITIATION QUEUE
7451  035312  000207                  60$:      RTS     PC              ;RETURN
7452
7453  035314  032765  040000  000020  E5$:      BIT     #WCE,P.CS2(R5)  ;CHECK IF WRITE CHECK ERROR
7454  035322  001002                            BNE     1$              ;YES, PROCESS WRITE CHECK ERROR
7455  035324  000137  035466                    JMP     E10$            ;NO, CONTINUE
7456
7457  035330  004037  031032          1$:       JSR     R0,PRI200       ;PRINT WRITE CHECK ERROR
7458  035334  062774                            ERR050
7459  035336  105765  000120                    TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR CORRECTION AND RETRY
7460  035342  001006                            BNE     5$              ;YES, INITIATE NEXT COMMAND
7461  035344  105265  000147                    INCB    P.RERD(R5)      ;INCREMENT RETRY COUNT
7462  035350  122765  000021  000147            CMPB    #21,P.RERD(R5)  ;CHECK IF 16 REREADS OCCURRED
7463  035356  101003                            BHI     10$             ;NO, START RECOVERY SEQUENCE
7464  035360  004737  031660          5$:       JSR     PC,UNRECV       ;INDICATE UNSUCCESSFUL RECOVERY
7465  035364  000207                            RTS     PC              ;RETURN
7466
7467  035366  016537  000024  001634  10$:      MOV     P.BAR(R5),RTYBA ;GET ADDRESS OF WORD IN ERROR
7468  035374  162737  000002  001634            SUB     #2,RTYBA        ;ADJUST ADDRESS
```

```
7469  035402  004737  035770                    JSR     PC,CALSEC       ;CALCULATE SECTOR, TRACK AND CYLINDER
7470  035406  116565  000123  000001            MOVB    P.RCMD(R5),P.CMND(R5) ;GET PREVIOUS COMMAND
7471  035414  013765  001640  000004            MOV     RTYSEC,P.SECT(R5) ;GET SECTOR AND TRACK
7472  035422  013765  001642  000002            MOV     RTYCYL,P.CYLN(R5) ;GET CYLINDER
7473  035430  013765  001634  000010            MOV     RTYBA,P.BALO(R5) ;GET BUS ADDRESS
7474  035436  013765  001636  000012            MOV     RTYWC,P.WC(R5)  ;GET WORD COUNT
7475  035444  005465  000012                    NEG     P.WC(R5)        ;MAKE IT NEGATIVE
7476  035450  052765  000200  000014            BIS     #W.WCK,P.PRST(R5) ;SET ISSUE WRITE BEFORE WRITE CHECK
7477  035456  004037  051564                    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER IN
7478  035462  001324                            CINITQ                  ;  COMMAND INITIATION QUEUE
7479  035464  000207                            RTS     PC              ;RETURN
7480
7481
7482  035466  032765  000004  000212  E10$:     BIT     #BIT2,P.ERR(R5) ;CHECK IF DATA COMPARE ERROR
7483  035474  001001                            BNE     2$              ;YES, CONTINUE PROCESSING ERROR
7484  035476  000000                            HALT                    ;  *** UNKNOWN ENTRY
7485  035500  005046                    2$:     CLR     -(SP)           ;MAKE ROOM ON STACK
7486  035502  116516  000217                    MOVB    P.ERAD(R5),(SP) ;GET WORD IN ERROR
7487  035506  006316                            ASL     (SP)            ;MULTIPLY BY 2
7488  035510  066516  000214                    ADD     P.ERHD(R5),(SP) ;CALCULATE ADDRESS OF MISCOMPARED WORD
7489  035514  012637  001634                    MOV     (SP)+,RTYBA     ;LOAD ADDRESS OF BAD WORD
7490  035520  004737  035770                    JSR     PC,CALSEC       ;CALCULATE CYLINDER, TRACK, AND SECTOR
7491  035524  032765  000002  000150            BIT     #BIT1,P.DSTT(R5) ;CHECK IF DATA TYPE ERROR HAS ALREADY OCCURRED
7492  035532  001037                            BNE     10$             ;YES, GO TO RETRY SEQUENCE
7493  035534  052765  000002  000150            BIS     #BIT1,P.DSTT(R5) ;SET DATA TYPE ERROR OCCURRED
7494  035542  100003                            BPL     3$              ;CHECK IF FIRST ERROR
7495  035544  052765  000001  000150            BIS     #BIT0,P.DSTT(R5) ;SET INITIAL DATA ERROR
7496  035552  004737  030156            3$:     JSR     PC,PRI000       ;PRINT HEADER
7497  035556  032777  020000  143354            BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
7498  035564  001013                            BNE     5$              ;YES, CHECK IF HALT ON ERROR
7499  035566  104401  063446                    TYPE    ,ERR062         ;TYPE DATA COMPARE ERROR
7500  035572  005765  000150                    TST     P.DSTT(R5)      ;CHECK IF FIRST ERROR
7501  035576  100402                            BMI     4$              ;YES, GO PRINT ERROR WORD
7502  035600  104401  064317                    TYPE    ,ERR202         ;TYPE ERROR DURING ERROR RECOVERY
7503  035604  004737  036202            4$:     JSR     PC,PRERWD       ;PRINT WORD IN ERROR
7504  035610  004737  030422                    JSR     PC,PRI002       ;PRINT CURRENTLY SUPPLIED COMMAND
7505  035614  042765  100200  000150    5$:     BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
7506  035622  004737  027620                    JSR     PC,PRIBUF       ;GO PRINT BUFFER
7507  035626  004737  031426                    JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
7508  035632  105765  000120            10$:    TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR CORRECTION AND RECOVERY
7509  035636  001006                            BNE     15$             ;YES, INDICATE UNSUCCESSFUL RECOVERY
7510  035640  105265  000147                    INCB    P.RERD(R5)      ;INCREMENT RETRY COUNT
7511  035644  122765  000043  000147            CMPB    #43,P.RERD(R5)  ;CHECK IF ERROR RECOVERY UNSUCCESSFUL
7512  035652  101003                            BHI     20$             ;NO, GO TO RECOVERY SEQUENCE
7513  035654  004737  031660            15$:    JSR     PC,UNRECY       ;INDICATE UNSUCCESSFUL RECOVERY
7514  035660  000207                            RTS     PC              ;RETURN
7515
```

```
 7516  035662  122765  000021  000147  20$:    CMPB    #21,P.RERD(R5)  ;CHECK IF ENTERING OFFSET
 7517  035670  001007                          BNE     25$             ;NO, DETERMINE IF TIME TO CHANGE OFFSET
 7518  035672  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
 7519  035674  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
 7520  035700  146437  001514  001513          BICB    INTMSK(R4),O.OVER ;RESET SEEK BEFORE DATA TRANSFER
 7521  035706  012604                          MOV     (SP)+,R4        ;RESTORE R4
 7522  035710  004037  037204          25$:    JSR     R0,DETOFF       ;DETERMINE IF OFFSET TO BE ISSUED
 7523  035714  035766                          50$                     ;OFFSET HAS BEEN ISSUED
 7524  035716  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;GET LAST ISSUED COMMAND
 7525  035724  013765  001640  000004          MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK ADDRESS
 7526  035732  013765  001642  000002          MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER ADDRESS
 7527  035740  013765  001634  000010          MOV     RTYBA,P.BALO(R5) ;LOAD BUS ADDRESS
 7528  035746  013765  001636  000012          MOV     RTYWC,P.WC(R5)  ;GET WORD COUNT
 7529  035754  005465  000012                  NEG     P.WC(R5)        ;MAKE IT NEGATIVE
 7530  035760  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7531  035764  001324                          CINITQ                  ;  INITIATION QUEUE
 7532  035766  000207                  50$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  CALCULATE CYLINDER, TRACK, AND SECTOR FOR RETRY
                                
035770  010146                  CALSEC: MOV     R1,-(SP)          ;STORE R1 ON STACK
035772  010346                          MOV     R3,-(SP)          ;STORE R3 ON STACK
035774  013703  001634                  MOV     RTYBA,R3          ;GET ADDRESS OF ERROR WORD
036000  166503  000010                  SUB     P.BALO(R5),R3     ;CALCULATE BYTES CORRECTLY TRANSFERRED
036004  042703  000777                  BIC     #777,R3           ;MAKE IT ON SECTOR BOUNDARY
036010  010337  001634                  MOV     R3,RTYBA          ;STORE CALCULATED BUS ADDRESS
036014  066537  000010  001634          ADD     P.BALO(R5),RTYBA  ;CALCULATE START OF SECTOR WITH ERROR
036022  016501  000012                  MOV     P.WC(R5),R1       ;GET INITIAL WORD COUNT
036026  005401                          NEG     R1                ;MAKE IT POSITIVE
036030  006203                          ASR     R3                ;DETERMINE WORDS BEFORE ERROR
036032  160301                          SUB     R3,R1             ;CALCULATE NUMBER OF WORDS LEFT
036034  022701  000400                  CMP     #256.,R1          ;CH3ECK IF LESS THAN OR EQUAL TO ONE SECTOR
036040  103002                          BHIS    5$                ;LOAD RETRY WORD COUNT
036042  012701  000400                  MOV     #256.,R1          ;NO, MAKE IT ONE SECTOR
036046  010137  001636          5$:     MOV     R1,RTYWC          ;LOAD RETY WORD COUNT
036052  116537  000005  001641          MOVB    P.TRCK(R5),RTYTRK ;STORE RETRY TRACK
036060  016537  000002  001642          MOV     P.CYLN(R5),RTYCYL ;STORE RETRY CYLINDER
036066  010346                          MOV     R3,-(SP)          ;STORE WORDS TRANSFERRED
036070  116616  000001                  MOVB    1(SP),(SP)        ;STORE NUMBER OF SECTORS
036074  105066  000001                  CLRB    1(SP)             ;CLEAR HIGH ORDER BYTE
036100  005046                          CLR     -(SP)             ;MAKE ROOM ON STACK
036102  116516  000004                  MOVB    P.SECT(R5),(SP)   ;GET INITIAL SECTOR
036106  062616                          ADD     (SP)+,(SP)        ;ADD SECTORS TRANSFERRED
036110  022716  000025          10$:    CMP     #21.,(SP)         ;CHECK IF SECTOR COUNT GREATER THAN 21
036114  103005                          BHIS    15$               ;NO, CHECK TRACK
036116  162716  000026                  SUB     #22.,(SP)         ;SUBTRACT 22 SECTORS/TRACK
036122  105237  001641                  INCB    RTYTRK            ;INCREMENT TRACK
036126  000770                          BR      10$               ;CHECK SECTOR COUNT
                                
036130  112637  001640          15$:    MOVB    (SP)+,RTYSEC      ;STORE RETRY SECTOR
036134  005046                          CLR     -(SP)             ;MAKE ROOM ON THE STACK
036136  113716  001641                  MOVE    RTYTRK,(SP)       ;STORE RETRY TRACK
036142  122716  000002          17$:    CMPE    #2,(SP)           ;CHECK TRACK GREATER THAN 2
036146  103005                          BHIS    20$               ;NO, RETURN
036150  162716  000003                  SUB     #3,(SP)           ;SUBTRACT 3 TRACKS/CYLINDER
036154  005237  001642                  INC     RTYCYL            ;INCREMENT CYLINDER
036160  000770                          BR      17$               ;CHECK TRACK COUNT
                                
036162  112637  001641          20$:    MOVB    (SP)+,RTYTRK      ;LOAD RETRY COUNT
036166  112737  000001  001644          MOVB    #1,RTYSCN         ;INDICATE ONE SECTOR IN RETRY COUNT
036174  012603                          MOV     (SP)+,R3          ;RESTORE R3
036176  012601                          MOV     (SP)+,R1          ;RESTORE R1
036200  000207                          RTS     PC                ;RETURN
```

```
                                        .SBTTL  PRINT FIRST ERROR WORD IN BUFFER
                                        
036202  010046                  PRERWD: MOV     R0,-(SP)        ;STORE R0 ON STACK
036204  010146                          MOV     R1,-(SP)        ;STORE R1 ON STACK
036206  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
036210  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
036212  104401  061571                  TYPE    ,ERR012         ;TYPE ERROR HEADER
036216  016503  000214                  MOV     P.ERHD(R5),R3   ;GET ADDRESS OF START OF SECTOR
036222  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK
036224  116516  000217                  MOVB    P.ERAD(R5),(SP) ;GET WORD IN ERROR OF SECTOR
036230  006316                          ASL     (SP)            ;MULTIPLY IT BY 2
036232  011601                          MOV     (SP),R1         ;STORE IT IN R1
036234  062603                          ADD     (SP)+,R3        ;GET ADDRESS OF BAD WORD
036236  010346                          MOV     R3,-(SP)        ;STORE ADDRESS FOR PRINT OUT
036240  004737  053224                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
036244  122765  000001  000217          CMPB    #1,P.ERAD(R5)   ;CHECK IF IN DESCRIPTOR AREA
036252  103403                          BLO     5$              ;NO, CALCULATE WORD
036254  104401  061113                  TYPE    ,QUESMK         ;TYPE QUESTION MARKS
036260  000436                          BR      10$             ;GO TYPE WORD READ
                                        
036262  016504  000214          5$:     MOV     P.ERHD(R5),R4   ;GET HEADER FOR PATTERN INDEX
036266  016400  000002                  MOV     2(R4),R0        ;GET NUMBER OF WORDS WRITTEN IN SECTOR
036272  042700  177400                  BIC     #177400,R0      ;KEEP LOW BYTE
036276  005200                          INC     R0              ;INCREMENT R0 TO MAKE OFFSET FROM START
                                                                ;  OF SECTOR
036300  126500  000217                  CMPB    P.ERAD(R5),R0   ;CHECK IF IN ZERO FILL AREA
036304  101402                          BLOS    7$              ;NO, CALCULATE ADDRESS OF
                                                                ;  WORD IN DATA PATTERN
036306  005046                          CLR     -(SP)           ;LOAD ZERO FOR PRINT OUT
036310  000414                          BR      9$              ;GO PRINT EXPECTED CONTENTS
                                        
036312  011404                  7$:     MOV     (R4),R4         ;GET DATA PATTERN NUMBER FOR INDEX
036314  042704  177760                  BIC     #177760,R4      ;MASK OFF INSIGNIFICANT BITS
036320  006304                          ASL     R4              ;MULTIPLY BY 2
036322  016404  004602                  MOV     PATTAB(R4),R4   ;GET PATTERN BASE
036326  162701  000004                  SUB     #4,R1           ;SUBTRACT 4 FROM SECTOR OFFSET
036332  042701  177700                  BIC     #177700,R1      ;MAKE IT MOD 32 WORDS
036336  060104                          ADD     R1,R4           ;CALCULATE PATTERN ADDRESS
036340  011446                          MOV     (R4),-(SP)      ;STORE EXPECTED CONTENTS FOR PRINT OUT
036342  104401  061126          9$:     TYPE    ,BLANKS         ;TYPE BLANKS
036346  004737  053224                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
036352  104401  061126                  TYPE    ,BLANKS         ;TYPE BLANKS
036356  011346                  10$:    MOV     (R3),-(SP)      ;STORE WORD READ
036360  004737  053224                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
036364  104401  061126                  TYPE    ,BLANKS         ;TYPE BLANKS
036370  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK
036372  116516  000217                  MOVB    P.ERAD(R5),(SP) ;STORE SECTOR WORD NUMBER FOR PRINT OUT
036376  004737  053224                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
036402  105765  000217                  TSTB    P.ERAD(R5)      ;CHECK IF ERROR IS IN PATTERN WORD
036406  001003                          BNE     15$             ;NO, PRINT PATTERN NUMBER
036410  104401  061113                  TYPE    ,QUESMK         ;TYPE QUESTION MARKS
036414  000410                          BR      20$             ;GO RESTORE REGISTERS
                                        
036416  104401  061126          15$:    TYPE    ,BLANKS         ;TYPE BLANKS
036422  017646  000214                  MOV     @P.ERHD(R5),-(SP) ;GET FIRST WORD OF SECTOR READ
036426  042716  177760                  BIC     #177760,(SP)    ;CLEAR INSIGNIFICANT BITS
```

```
 7634  036432  004737  053224            JSR     PC,BINOCT       ;CONVERT TO OCTAL
 7635  036436  104401  001165     20$:   TYPE    ,$CRLF          ;TYPE <CR><LF>
 7636  036442  012604                    MOV     (SP)+,R4        ;RESTORE R4
 7637  036444  012603                    MOV     (SP)+,R3        ;RESTORE R3
 7638  036446  012601                    MOV     (SP)+,R1        ;RESTORE R1
 7639  036450  012600                    MOV     (SP)+,R0        ;RESTORE R0
 7640  036452  000207                    RTS     PC              ;RETURN
```

```
                                        .SBTTL  SUCCESSFUL RECOVERY CLEAN UP

036454  032765  000001  000150  SURECY: BIT     #BIT0,P.DSTT(R5) ;CHECK IF DATA TYPE ERROR
036462  001466                          BEQ     10$              ;NO, CLEAN UP FOR NEXT COMMAND
036464  105765  000147                  TSTB    P.RERD(R5)       ;CHECK IF FIRST ERROR
036470  001003                          BNE     5$               ;NO, CHECK IF OFFSET
036472  005265  000176                  INC     P.SECC(R5)       ;INCREMENT NO. OF ECC RECOVERABLE ERRORS
036476  000426                          BR      9$               ;GO CLEAN UP FOR NEXT COMMAND

036500  122765  000020  000147  5$:     CMPB    #20,P.RERD(R5) ;CHECK IF OFFSET
036506  103403                          BLO     8$             ;YES, INCREMENT OFFSET RECOVERABLE
036510  005265  000172                  INC     P.SRRD(R5)     ;INCREMENT REREAD RECOVERABLE
036514  000417                          BR      9$             ;GO CLEAN UP FOR NEXT COMMAND

036516  005265  000174          8$:     INC     P.SOFF(R5)     ;INCREMENT OFFSET RECOVERABLE
036522  032777  020000  142410          BIT     #SW13,@SWR     ;CHECK IF INHIBIT PRINT OUT
036530  001011                          BNE     9$             ;YES, CLEAN UP FOR NEXT COMMAND
036532  104401  064150                  TYPE    ,ERR108        ;TYPE "OFFSET"
036536  005046                          CLR     -(SP)          ;MAKE ROOM ON STACK
036540  116516  000006                  MOVB    P.OFST(R5),(SP) ;STORE OFFSET FOR PRINT OUT
036544  004737  053224                  JSR     PC,BINOCT      ;CONVERT TO OCTAL
036550  104401  001165                  TYPE    ,$CRLF         ;TYPE <CR><LF>
036554  105037  001644          9$:     CLRB    RTYSCN         ;CLEAR SECTOR RETRY FLAG
036560  016546  000176                  MOV     P.SECC(R5),-(SP) ;STORE NUMBER OF ECC RECOVERABLE ERRORS
036564  066516  000172                  ADD     P.SRRD(R5),(SP) ;ADD NUMBER OF REREAD RECOVERABLE ERRORS
036570  066516  000174                  ADD     P.SOFF(R5),(SP) ;ADD NUMBER OF OFFSET RECOVERABLE ERRORS
036574  022665  000100                  CMP     (SP)+,P.CERT(R5) ;CHECK IF SOFT ERROR RATE EXCEEDED
036600  101417                          BLOS    10$              ;NO, GO ISSUE NEXT COMMAND
036602  032777  040020  142330          BIT     #SW4!SW14,@SWR   ;CHECK IF INHIBIT AUTOMATIC DRIVE DROPPING
036610  001013                          BNE     10$              ;YES, GO ISSUE NEXT COMMAND
036612  004737  023020                  JSR     PC,BUFREL        ;GO RELEASE BUFFER
036616  104401  064445                  TYPE    ,ERR205          ;TYPE "ERROR THRESHOLD EXCEEDED"
036622  004737  013162                  JSR     PC,DROP          ;GO DROP DRIVE
036626  005037  001566                  CLR     ERRPRO           ;CLEAR ERROR CURRENTLY BEING PROCESSED
036632  004737  022520                  JSR     PC,GETBUF        ;GET BUFFER FOR NEXT COMMAND
036636  000207                          RTS     PC               ;RETURN

036640  032777  020000  142272  10$:    BIT     #SW13,@SWR       ;CHECK IF INHIBIT PRINT OUT
036646  001002                          BNE     15$              ;YES, CHECK IF READ PACK SERIAL NUMBER
036650  104401  064355                  TYPE    ,ERR203          ;TYPE RECOVERABLE ERROR
036654  105037  001644          15$:    CLRB    RTYSCN           ;CLEAR SECTOR RETRY COUNT
036660  005065  000146                  CLR     P.RECT(R5)       ;CLEAR ERROR FLAGS AND RETRY COUNTS
036664  005065  000150                  CLR     P.DSTT(R5)
036670  005065  000212                  CLR     P.ERR(R5)
036674  005037  001566                  CLR     ERRPRO           ;CLEAR ERROR CURRENTLY BEING PROCESSED
036700  010446                          MOV     R4,-(SP)         ;STORE R4 ON STACK
036702  116504  000000                  MOVB    P.DRVN(R5),R4    ;STORE DRIVE NUMBER FOR INDEX
036706  156437  001514  001513          BISB    INTMSK(R4),O.OVER ;ALLOW OVERLAPPED OPERATIONS ON THIS DRIVE
036714  012604                          MOV     (SP)+,R4         ;RESTORE R4
036716  105765  000210                  TSTB    P.ASSN(R5)       ;CHECK IF IN ASSIGNMENT SEQUENCE
036722  001025                          BNE     20$              ;YES, TERMINATE SEQUENCE
036724  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK TO PREVIOUS CYLINDER
036732  001410                          BEQ     16$              ;YES, GO ISSUE RELEASE
036734  122765  000140  000001          CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE
036742  001013                          BNE     17$              ;NO, CARRY OUT NORMAL RECOVERY
036744  004037  051564                  JSR     R0,Q.PUSH        ;ENQUEUE PARAMETER BLOCK IN DRIVE
```

```
 7697  036750  001314                          AVAILQ                  ;  AVAILIABLE QUEUE
 7698  036752  000207                          RTS     PC              ;RETURN
 7699
 7700  036754  112765  000140  000001  16$:    MOVB    #RELEAS,P.CMND(R5) ;ISSUE RELEASE COMMAND
 7701  036762  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7702  036766  001324                          CINITQ                  ;  INITIATION QUEUE
 7703  036770  000207                          RTS     PC              ;RETURN
 7704
 7705  036772  000137  024774          17$:    JMP     NORM1           ;CARRY OUT NORMAL DATA TRANSFER TERMINATION
 7706
 7707                                  ;       CHECK IF ANY OF THE FOLLOWING ASSIGNMENT STAGES
 7708                                  ;               READ DRIVE STATUS
 7709                                  ;               START SPINDLE
 7710                                  ;               SET VOLUME VALID
 7711                                  ;               RECAL
 7712                                  ;               READ PACK SERIAL NUMBER
 7713  036776  132765  000117  000210  20$:    BITB    #BIT0!BIT1!BIT2!BIT3!BIT6,P.ASSN(R5)
 7714  037004  001402                          BEQ     25$             ;NO, PROCESS WRITE PACK
 7715  037006  000137  013662                  JMP     ASNORM          ;PROCESS NORMAL TERMINATION
 7716
 7717  037012  032765  002000  000014  25$:    BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
 7718  037020  001030                          BNE     30$             ;YES, GO DROP DRIVE
 7719  037022  105065  000211                  CLRB    P.SEEK(R5)      ;CLEAR SEEK TO PREVIOUS LOCATION
 7720  037026  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE LAST WRITE/WRITE CHECK COMMAND
 7721  037034  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;SET ISSUE WRITE BEFORE WRITE CHECK
 7722  037042  016565  000126  000004          MOV     P.RSEC(R5),P.SECT(R5) ;LOAD PREVIOUS TRACK AND SECTOR
 7723  037050  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5) ;LOAD PREVIOUS CYLINDER
 7724  037056  016565  000132  000010          MOV     P.RBAL(R5),P.BALO(R5) ;LOAD PREVIOUS BUS ADDRESS
 7725  037064  016565  000130  000012          MOV     P.RWC(R5),P.WC(R5) ;LOAD PREVIOUS WORD COUNT
 7726  037072  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7727  037076  001324                          CINITQ                  ;  INITIATION QUEUE
 7728  037100  000207                          RTS     PC              ;RETURN
 7729
 7730  037102  004737  023020          30$:    JSR     PC,BUFREL       ;GO RELEASE BUFFER
 7731  037106  104401  056656                  TYPE    ,OPR008         ;TYPE 'OPERATOR INITIATED DROP DRIVE
 7732                                                                  ;  DURING PACK WRITING'
 7733  037112  004737  013162                  JSR     PC,DROP         ;GO DROP DRIVE
 7734  037116  004737  022520                  JSR     PC,GETBUF       ;GET BUFFER FOR NEXT COMMAND
 7735  037122  000207                          RTS     PC              ;RETURN
```

```
 7736                                   .SBTTL  GET CURRENT STATUS INFORMATION
 7737
 7738  037124  010446                   GETCUR: MOV     R4,-(SP)          ;STORE R4 ON STACK
 7739  037126  010346                           MOV     R3,-(SP)          ;STORE R3 ON STACK
 7740  037130  010504                           MOV     R5,R4             ;LOAD START OF CURRENT PARRAMETER BLOCK
 7741  037132  012703  004516                   MOV     #PARM10,R3        ;LOAD START OF RETRY PARAMETER BLOCK
 7742
 7743  037136  012423                   5$:     MOV     (R4)+,(R3)+       ;COPY PARAMTER BLOCK
 7744  037140  022703  004602                   CMP     #PARM10+P.EPAT+2,R3 ;CHECK IF FINISHED
 7745  037144  001374                           BNE     5$                ;NO, GET NEXT WORD
 7746  037146  012603                           MOV     (SP)+,R3          ;RESTORE R3
 7747  037150  012604                           MOV     (SP)+,R4          ;RESTORE R4
 7748  037152  000207                           RTS     PC                ;RETURN
 7749
 7750                                   .SBTTL  RESTORE STATUS
 7751
 7752  037154  010446                   RSTAT:  MOV     R4,-(SP)          ;STORE R4 ON STACK
 7753  037156  010346                           MOV     R3,-(SP)          ;STORE R3 ON STACK
 7754  037160  010504                           MOV     R5,R4             ;LOAD START OF CURRENT PARRAMETER BLOCK
 7755  037162  012703  004516                   MOV     #PARM10,R3        ;LOAD START OF RETRY PARAMETER BLOCK
 7756
 7757  037166  012324                   5$:     MOV     (R3)+,(R4)+       ;COPY PARAMTER BLOCK
 7758  037170  022703  004602                   CMP     #PARM10+P.EPAT+2,R3 ;CHECK IF FINISHED
 7759  037174  001374                           BNE     5$                ;NO, GET NEXT WORD
 7760  037176  012603                           MOV     (SP)+,R3          ;RESTORE R3
 7761  037200  012604                           MOV     (SP)+,R4          ;RESTORE R4
 7762  037202  000207                           RTS     PC                ;RETURN
```

```
 7763                                   .SBTTL  DETERMINE OFFSET FOR RETRY
 7764
 7765  037204  122765  000021  000147  DETOFF: CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET = +400 MICRO-INCHES
 7766  037212  001004                          BNE     10$             ;NO, CONTINUE
 7767  037214  112765  000020  000006          MOVB    #OFFP4,P.OFST(R5) ;LOAD OFFSET VALUE = +400 MICRO-INCHES
 7768  037222  000447                          BR      15$             ;ISSUE OFFSET
 7769
 7770  037224  122765  000024  000147  10$:    CMPB    #24,P.RERD(R5)  ;CHECK IF OFFSET = -400 MICRO-INCHES
 7771  037232  001004                          BNE     11$             ;NO, CONTINUE
 7772  037234  112765  000220  000006          MOVB    #OFFM4,P.OFST(R5) ;LOAD OFFSET VALUE = -400 MICRO-INCHES
 7773  037242  000437                          BR      15$             ;ISSUE OFFSET
 7774
 7775  037244  122765  000027  000147  11$:    CMPB    #27,P.RERD(R5)  ;CHECK IF OFFSET = +800 MICRO-INCHES
 7776  037252  001004                          BNE     12$             ;NO, CONTINUE
 7777  037254  112765  000040  000006          MOVB    #OFFP8,P.OFST(R5) ;LOAD OFFSET VALUE = +800 MICRO-INCHES
 7778  037262  000427                          BR      15$             ;ISSUE OFFSET
 7779
 7780  037264  122765  000032  000147  12$:    CMPB    #32,P.RERD(R5)  ;CHECK IF OFFSET = -800 MICRO-INCHES
 7781  037272  001004                          BNE     13$             ;NO, CONTINUE
 7782  037274  112765  000240  000006          MOVB    #OFFM8,P.OFST(R5) ;LOAD OFFSET VALUE = -800 MICRO-INCHES
 7783  037302  000417                          BR      15$             ;ISSUE OFFSET
 7784
 7785  037304  122765  000035  000147  13$:    CMPB    #35,P.RERD(R5)  ;CHECK IF OFFSET = +1200 MICRO-INCHES
 7786  037312  001004                          BNE     14$             ;NO, CONTINUE
 7787  037314  112765  000060  000006          MOVB    #OFFP12,P.OFST(R5) ;LOAD OFFSET VALUE = +1200 MICRO-INCHES
 7788  037322  000407                          BR      15$             ;ISSUE OFFSET
 7789
 7790  037324  122765  000040  000147  14$:    CMPB    #40,P.RERD(R5)  ;CHECK IF OFFSET = -1200 MICRO-INCHES
 7791  037332  001013                          BNE     30$             ;NO, DO REREAD
 7792  037334  112765  000260  000006          MOVB    #OFFM12,P.OFST(R5) ;LOAD OFFSET VALUE = -1200 MICRO-INCHES
 7793  037342  112765  000115  000001  15$:    MOVB    #OFFSET,P.CMND(R5) ;LOAD OFFSET COMMAND
 7794  037350  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7795  037354  001324                          CINITQ                  ;  INITIATION QUEUE
 7796  037356  011000                          MOV     (R0),R0         ;LOAD OFFSET ISSUED RETURN
 7797  037360  000200                          RTS     R0              ;RETURN
 7798
 7799  037362  005720                  30$:    TST     (R0)+           ;ADJUST RETURN FOR NO OFFSET
 7800  037364  000200                          RTS     R0              ;RETURN
```

```
 7801                                   .SBTTL  CALCULATE RETRY WORD COUNT
 7802
 7803  037366  013746  001642  CALWC:   MOV     RTYCYL,-(SP)      ;STORE RETRY CYLINDER
 7804  037372  166516  000002           SUB     P.CYLN(R5),(SP)   ;SUBTRACT INITIAL CYLINDER
 7805  037376  011646                   MOV     (SP),-(SP)        ;STORE CYLINDER DIFFERENCE
 7806  037400  006316                   ASL     (SP)              ;MULTIPLY DIFFERENCE BY 2
 7807  037402  062616                   ADD     (SP)+,(SP)        ;MULTIPLY DIFFERENCE BY 3
 7808  037404  005046                   CLR     -(SP)             ;MAKE ROOM ON STACK
 7809  037406  113716  001641           MOVB    RTYTRK,(SP)       ;GET RETRY TRACK
 7810  037412  062616                   ADD     (SP)+,(SP)        ;ADD RETRY TRACK
 7811  037414  005046                   CLR     -(SP)             ;MAKE ROOM ON THE STACK
 7812  037416  116516  000005           MOVB    P.TRCK(R5),(SP)   ;GET INITIAL TRACK
 7813  037422  162616                   SUB     (SP)+,(SP)        ;SUBTRACT INITIAL TRACK
 7814  037424  012746  000026           MOV     #22.,-(SP)        ;MULTIPLY TRACK BY 22
 7815  037430  004737  055210           JSR     PC,$MULT
 7816  037434  012616                   MOV     (SP)+,(SP)        ;THROW AWAY MOST SIGNIFICANT BITS
 7817  037436  005046                   CLR     -(SP)             ;MAKE ROOM ON THE STACK
 7818  037440  113716  001640           MOVB    RTYSEC,(SP)       ;GET RETRY SECTOR
 7819  037444  062616                   ADD     (SP)+,(SP)        ;ADD PRESENT SECTOR
 7820  037446  005046                   CLR     -(SP)             ;MAKE ROOM ON STACK
 7821  037450  116516  000004           MOVB    P.SECT(R5),(SP)   ;GET INITIAL SECTOR
 7822  037454  162616                   SUB     (SP)+,(SP)        ;CALCULATE NUMBER OF SECTORS UNTIL ERROR
 7823  037456  111666  000001           MOVB    (SP),1(SP)        ;MULTIPLY BY 256 WORDS/SECTOR
 7824  037462  105016                   CLRB    (SP)
 7825  037464  161637  001636           SUB     (SP),RTYWC        ;SUBTRACT FROM INITIAL WORD COUNT
 7826  037470  006316                   ASL     (SP)              ;DETERMINE NUMBER OF BYTES
 7827  037472  062637  001634           ADD     (SP)+,RTYBA       ;CALCULATE STARTING BUFFER ADDRESS
 7828  037476  000207                   RTS     PC                ;RETURN
```

```
                                        .SBTTL  RECOVERABLE CONTROLLER ERROR TERMINATION

037500  012037  037522         CONERR: MOV     (R0)+,2$        ;STORE ERROR MESSAGE ADDRESS
037504  004737  030156                 JSR     PC,PRI000       ;PRINTER HEADER
037510  032777  020000  141422         BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
037516  001032                       BNE     10$             ;YES, CHECK IF HALT ON ERROR
037520  104401                       TYPE                    ;TYPE ERROR MESSAGE
037522  000000                 2$:   .WORD   0
037524  104401  001165               TYPE    ,$CRLF          ;TYPE <CR><LF>
037530  004737  026252               JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
037534  004737  030422               JSR     PC,PRI002       ;PRINT CURRENT EXECUTED COMMAND
037540  022737  063526  037522       CMP     #ERR066,2$      ;CHECK IF ECC LOGIC ERROR
037546  001016                       BNE     10$             ;NO, CHECK IF HALT ON ERROR
037550  104401  063546               TYPE    ,ERR067         ;TYPE "RKECPT RKECPS"
037554  016546  000062               MOV     P.EPAT(R5),-(SP) ;STORE ECC PATTERN FOR PRINT OUT
037560  004737  053224               JSR     PC,BINOCT       ;CONVERT TO OCTAL
037564  104401  061126               TYPE    ,BLANKS         ;TYPE 2 SPACES
037570  016546  000060               MOV     P.EPOS(R5),-(SP) ;STORE ECC POSITION FOR PRINT OUT
037574  004737  053224               JSR     PC,BINOCT       ;CONVERT TO OCTAL
037600  104401  001165               TYPE    ,$CRLF          ;TYPE <CR><LF>
037604  004737  031426         10$:  JSR     PC,HLT000       ; HECK IF HALT ON ERROR
037610  005237  001630               INC     CNTCNT          ;INCREMENT CONTROLLER ERROR COUNT
037614  022737  000012  001630       CMP     #T.CONT,CNTCNT  ;CHECK IF THRESHOLD EXCEEDED
037622  103006                       BHIS    15$             ;NO, CONTINUE
037624  104401  061131               TYPE    ,ERR000         ;TYPE "FATAL ERROR"
037630  104401  061470               TYPE    ,ERR008         ;TYPE "CONTROLLER ERROR THRESHOLD EXCEEDED"
037634  000000                       HALT
037636  000776                       BR      .-2

037640  105765  000150         15$:  TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
037644  100005                       BPL     20$             ;NO, REQUEUE COMMAND
037646  042765  100200  000150       BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR ERROR ENQUEUED AND FIRST ERROR
037654  005037  001566               CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
037660  004037  051564         20$:  JSR     R0,Q.PUSH       ;REQUEUE COMMAND IN COMMAND
037664  001324                       CINITQ                  ;  INITIATION QUEUE
037666  000200                       RTS     R0              ;RETURN
```

```
                                        .SBTTL  TEST FOR DROP DRIVE ERRORS

037670  032765  000030  000036  TSTDRP: BIT     #ACLO!SPDLSS,P.DS(R5) ;CHECK IF AC LOW OR SPEED LOSS
037676  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
037700  032765  046064  000034        BIT     #UNS!IDAE!WLE!DTYE!FMTE!NXF,P.ER(R5)

037706  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
037710  032765  006010  000212        BIT     #BIT3!BIT10!BIT11,P.ERR(R5)

037716  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
037720  032765  004070  000014        BIT     #DRVHRD!DRVDSC!UEXATT!NODSC,P.PRST(R5)

037726  001116                  BNE     32$             ;YES REPORT ERROR
037730  005720                  TST     (R0)+           ;GET NORMAL RETURN ADDRESS
037732  000200                  RTS     R0              ;RETURN

037734  004737  030156          10$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
037740  032765  000010  000036        BIT     #ACLO,P.DS(R5)  ;CHECK IF AC LOW
037746  001403                  BEQ     11$             ;NO, CONTINUE
037750  104401  062501          TYPE    ,ERR033         ;TYPE "AC LOW"
037754  000472                  BR      30$             ;GO GET ERROR QUALIFICATION

037756  104401  062510          11$:    TYPE    ,ERR034         ;TYPE "SPEED LOSS"
037762  000467                  BR      30$             ;GET ERROR QUALIFICATION

037764  004737  030156          12$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
037770  032765  040000  000034        BIT     #UNS,P.ER(R5)   ;CHECK IF DRIVE UNSAFE
037776  001403                  BEQ     13$             ;NO, CONTINUE
040000  104401  063220          TYPE    ,ERR056         ;TYPE "DRIVE UNSAFE"
040004  000456                  BR      30$             ;GET ERROR QUALIFICATION

040006  032765  000040  000034  13$:    BIT     #DTYE,P.ER(R5)  ;CHECK IF DRIVE TYPE ERROR
040014  001403                  BEQ     14$             ;NO, CONTINUE
040016  104401  062523          TYPE    ,ERR035         ;TYPE "DRIVE TYPE ERROR"
040022  000447                  BR      30$             ;GET ERROR QUALIFIER BITS

040024  032765  000020  000034  14$:    BIT     #FMTE,P.ER(R5)  ;CHECK IF FORMAT ERROR
040032  001403                  BEQ     15$             ;NO, CONTINUE
040034  104401  062540          TYPE    ,ERR036         ;TYPE FORMAT ERROR
```

```
 7921  040040  000440                          BR      30$             ;GET ERROR QUAILIFER BITS
 7922
 7923  040042  032765  000004  000034  15$:    BIT     #NXF,P.ER(R5)   ;CHECK IF NON-EXECUTABLE DRIVE FUNCTION
 7924  040050  001403                          BEQ     16$             ;NO, CONTINUE
 7925  040052  104401  062553                  TYPE    ,ERR037         ;TYPE ILLEGAL DRIVE FUNCTION
 7926  040056  000431                          BR      30$             ;GET ERROR QUALIFIER BITS
 7927
 7928  040060  032765  002000  000034  16$:    BIT     #IDAE,P.ER(R5)  ;CHECK IF ILLEGAL DISK ADDRESS
 7929  040066  001403                          BEQ     17$             ;NO, CONTINUE
 7930  040070  104401  062450                  TYPE    ,ERR031         ;TYPE "ILLEGAL DISK ADDRESS"
 7931  040074  000422                          BR      30$             ;GET ERROR QUALIFER BITS
 7932
 7933  040076  104401  062464          17$:    TYPE    ,ERR032         ;TYPE 'WRITE LOCK ERROR"
 7934  040102  000417                          BR      30$             ;GET ERROR QUALIFIER BITS
 7935
 7936  040104  004737  030156          18$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
 7937  040110  032765  002000  000212          BIT     #BIT10,P.ERR(R5) ;CHECK IF DRIVE NOT READY AFTER START SPINDLE
 7938  040116  001403                          BEQ     19$             ;NO, CONTINUE
 7939  040120  104401  063112                  TYPE    ,ERR053+1       ;PRINT 'DRIVE NOT READY AFTER START SPINDLE"
 7940  040124  000406                          BR      30$             ;GET ERROR QUALIFIER BITS
 7941
 7942  040126  032765  004000  000212  19$:    BIT     #BIT11,P.ERR(R5) ;CHECK IF VOLUME VALID DID NOT SET
 7943                                                                  ;  AFTER PACK ACKNOWLEDGE
 7944  040134  001404                          BEQ     31$             ;NO, CONTINUE
 7945  040136  104401  063153                  TYPE    ,ERR054+1       ;PRINT "VOLUME VAID NOT SET AFTER
 7946                                                                  ;  PACK ACKNOWLEDGE INDSTRUCTION"
 7947  040142  104401  001165          30$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
 7948  040146  032765  000010  000212  31$:    BIT     #BIT3,P.ERR(R5) ;CHECK IF ERROR NOT INDICATED BY FAULT
 7949  040154  001405                          BEQ     42$             ;NO, CHECK OTHER QUALIFIER BITS
 7950  040156  104401  063233                  TYPE    ,ERR057         ;TYPE 'ERROR NOT INDICATED BY FAULT"
 7951  040162  000402                          BR      42$             ;CHECK OTHER QUALIFIER BITS
 7952
 7953  040164  004737  030156          32$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
 7954  040170  032765  000020  000014  42$:    BIT     #DRVHRD,P.PRST(R5) ;CHECK IF HARD DRIVE ERROR
 7955  040176  001402                          BEQ     43$             ;NO, CHECK OTHER QUALIFIER BITS
 7956  040200  104401  062321                  TYPE    ,ERR027         ;TYPE 'DRIVE HARD ERROR"
 7957  040204  032765  004000  000014  43$:    BIT     #NODSC,P.PRST(R5) ;CHECK IF ATTENTION BUT
 7958                                                                  ;  NO DRIVE STATUS CHANGE OR FAULT
 7959  040212  001402                          BEQ     44$             ;NO, CHECK OTHER QUALIFIER BITS
 7960  040214  104401  063274                  TYPE    ,ERR058         ;TYPE "ATTN BUT NO DSC OR FAULT"
 7961  040220  032765  000040  000014  44$:    BIT     #DRVDSC,P.PRST(R5) ;CHECK IF DRIVE STATUS CHANGE DID
 7962                                                                  ;  NOT CLEAR
 7963  040226  001402                          BEQ     45$             ;NO, CHECK OTHER ERROR QUALIFIER BITS
 7964  040230  104401  062340                  TYPE    ,ERR028         ;TYPE 'DRIVE STATUS CHANGE DID NOT CLEAR"
 7965  040234  032765  000010  000014  45$:    BIT     #UEXATT,P.PRST(R5) ;CHECK IF UNEXPECTED ATTENTION FROM
 7966                                                                  ;  DRIVE COMMAND
 7967  040242  001402                          BEQ     46$             ;NO, CHECK OTHER QUALIFIER BITS
 7968  040244  104401  062430                  TYPE    ,ERR030         ;TYPE 'UNEXPECTED ATTN FROM DRIVE COMMAND"
 7969  040250  005265  000206          46$:    INC     P.NER(R5)       ;INCREMENT OTHER ERROR COUNT
 7970  040254  032777  020000  140656          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
 7971  040262  001015                          BNE     60$             ;YES, CHECK IF HALT ON ERROR
 7972  040264  005765  000150                  TST     P.DSTT(R5)      ;CHECK IF ERROR DURING ERROR RECOVERY
 7973  040270  100402                          BMI     51$             ;NO, PRINT STATUS
 7974  040272  104401  064317                  TYPE    ,ERR202         ;TYPE 'ERROR DURING ERROR RECOVERY"
 7975  040276  004737  026252          51$:    JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
 7976  040302  004737  026076                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
```

```
 7977  040306  004737  030422                 JSR     PC,PRI002       ;PRINT CURRENT COMMAND
 7978  040312  004737  030244                 JSR     PC,PRI001       ;PRINT PREVIOUS COMMAND
 7979  040316  004737  031426          60$:   JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
 7980  040322  032777  000100  140610         BIT     #SW6,@SWR       ;CHECK FOR SWITCH 6
 7981  040330  001431                         BEQ     63$             ;NO DROP DRIVE
 7982  040332  032765  000030  000036         BIT     #ACLO!SPDLSS,P.DS(R5) ;CHECK IF UNSAFE, ACLOW AND SPEED LOSS
 7983  040340  001004                         BNE     61$
 7984  040342  032765  040000  000034         BIT     #UNS,P.ER(R5)
 7985  040350  001421                         BEQ     63$
 7986  040352  105765  000120          61$:   TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
 7987  040356  001012                         BNE     62$
 7988  040360  105265  000146                 INCB    P.RECT(R5)      ;INCREMENT ERROR RETRY COUNT
 7989  040364  122765  000010  000146         CMPB    #8.,P.RECT(R5)  ;CHECK IF RETRY THRESHOLD EXCEEDED
 7990  040372  103404                         BLO     62$             ;YES, INDICATE UNSUCCESSFUL RECOVERY
 7991  040374  004037  051564                 JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7992  040400  001324                         CINITQ                  ;  INITIATION QUEUE
 7993  040402  000402                         BR      57$             ;RETURN
 7994
 7995  040404  004737  031660          62$:   JSR     PC,UNRECY       ;INDICATE UNSUCCESSFUL RECOVERY
 7996  040410  011000                  57$:   MOV     (R0),R0
 7997  040412  000200                         RTS     R0
 7998
 7999  040414  004737  023020          63$:   JSR     PC,BUFREL       ;RELEASE BUFFER
 8000  040420  004737  013162                 JSR     PC,DROP         ;DROP DRIVE
 8001  040424  004737  022520                 JSR     PC,GETBUF       ;GET NEW BUFFER
 8002  040430  005037  001566                 CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
 8003  040434  011000                         MOV     (R0),R0         ;LOAD ERROR RETURN
 8004  040436  000200                         RTS     R0              ;RETURN
```

```
                                .SBTTL  ERROR RECOVERY SEQUENCE NORMAL RETURN
                        
040440  032765  004000  000150  ERCVY1: BIT     #BIT11,P.DSTT(R5) ;CHECK IF RECALIBRATE ISSUED
040446  001415                          BEQ     EC3$              ;NO, CONTINUE
040450  042765  004000  000150          BIC     #BIT11,P.DSTT(R5) ;CLEAR RECALIBRATE ISSUED
040456  052765  010000  000150  15$:     BIS     #BIT12,P.DSTT(R5) ;SET SEEK ISSUED
040464  112765  000117  000001          MOVB    #SEEK,P.CMND(R5) ;ISSUE SEEK INSTRUCTION
040472  004037  051564                  JSR     R0,Q.PUSH         ;PUT PARAMETER BLOCK IN COMMAND
040476  001324                          CINITQ                    ;  INITIATION QUEUE
040500  000207                          RTS     PC                ;RETURN
                        
040502  032765  010000  000150  EC3$:   BIT     #BIT12,P.DSTT(R5) ;CHECK IF SEEK ISSUED
040510  001427                          BEQ     10$               ;NO, CONTINUE
040512  042765  010000  000150          BIC     #BIT12,P.DSTT(R5) ;CLEAR SEEK ISSUED
040520  105765  000211                  TSTB    P.SEEK(R5)        ;CHECK IF SEEK TO PREVIOUS POSITION
040524  001403                          BEQ     5$                ;NO, PROCESS DATA TRANSFER
040526  004737  036454                  JSR     PC,SURECY         ;INDICATE SUCCESSFUL RECOVERY
040532  000207                          RTS     PC                ;RETURN
                        
040534  122765  000021  000147  5$:     CMPB    #21,P.RERD(R5)    ;CHECK IF OFFSET
040542  101053                          BHI     20$               ;NO, REISSUE COMMAND
040544  052765  020000  000150          BIS     #BIT13,P.DSTT(R5) ;SET OFFSET ISSUED
040552  112765  000115  000001          MOVB    #OFFSET,P.CMND(R5) ;ISSUE OFFSET
040560  004037  051564                  JSR     R0,Q.PUSH         ;PUT PARAMETER BLOCK IN COMMAND
040564  001324                          CINITQ                    ;  INITIATION QUEUE
040566  000207                          RTS     PC                ;RETURN
                        
040570  032765  020000  000150  10$:    BIT     #BIT13,P.DSTT(R5) ;CHECK IF DO PREVIOUS OFFSET
040576  001453                          BEQ     EC4$              ;NO, CONTINUE
040600  042765  020000  000150          BIC     #BIT13,P.DSTT(R5) ;CLEAR OFFSET ISSUED
040606  122765  000021  000147          CMPB    #21,P.RERD(R5)    ;CHECK IF OFFSET +400 MICRO-INCHES
040614  001424                          BEQ     15$               ;YES, GO ISSUE READ
040616  122765  000024  000147          CMPB    #24,P.RERD(R5)    ;CHECK IF OFFSET -400 MICRO-INCHES
040624  001420                          BEQ     15$               ;YES, GO ISSUE READ
040626  122765  000027  000147          CMPB    #27,P.RERD(R5)    ;CHECK IF OFFSET +800 MICRO-INCHES
040634  001414                          BEQ     15$               ;YES, GO ISSUE READ
040636  122765  000032  000147          CMPB    #32,P.RERD(R5)    ;CHECK IF OFFSET -800 MICRO-INCHES
040644  001410                          BEQ     15$               ;YES, GO ISSUE READ
040646  122765  000035  000147          CMPB    #35,P.RERD(R5)    ;CHECK IF OFFSET +1200 MICRO-INCHES
040654  001404                          BEQ     15$               ;YES, GO ISSUE READ
040656  122765  000040  000147          CMPB    #40,P.RERD(R5)    ;CHECK IF OFFSET -1200 MICRO-INCHES
040664  001002                          BNE     20$               ;NO, DO NOT INCREMENT REREAD COUNT
040666  105265  000147          15$:    INCB    P.RERD(R5)        ;INCREMENT REREAD COUNT
040672  116565  000123  000001  20$:    MOVB    P.RCMD(R5),P.CMND(R5) ;LOAD COMMAND
040700  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
040706  001003                          BNE     25$               ;NO, ISSUE COMMAND
040710  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;YES, ISSUE WRITE BEFOR WRITE CHECK
040716  004037  051564          25$:    JSR     R0,Q.PUSH         ;ENQUEUE PARAMETER BLOCK IN COMMAND
040722  001324                          CINITQ                    ;  INITIATION QUEUE
040724  000207                          RTS     PC                ;RETURN
                        
040726  032765  000100  000150  EC4$:   BIT     #BIT6,P.DSTT(R5) ;CHECK IF WAITING FOR READ STATUS
                                                                  ; FOR POSITIONING TIME OUT
040734  001436                          BEQ     EC5$              ;NO, CONTINUE
040736  042765  000100  000150          BIC     #BIT6,P.DSTT(R5) ;CLEAR WAITING FOR READ STATUS
040744  113765  001666  000001          MOVB    POSCMD,P.CMND(R5) ;RESTORE COMMAND OF ERROR
```

```
 8061  040752  032765  000200  000036          BIT     #DRDY,P.DS(R5)  ;CHECK IF DRIVE READY
 8062  040760  001004                          BNE     10$             ;YES, REISSUE COMMAND
 8063  040762  004037  030642                  JSR     R0,PRI100       ;PRINT DRIVE TIME OUT DURING POSITIONING
 8064  040766  063033                          ERR052                  ;  AND DROP DRIVE
 8065  040770  000207                  5$:     RTS     PC              ;RETURN
 8066
 8067  040772  004037  031032          10$:    JSR     R0,PRI200       ;PRINT DRIVE TIME OUT DURING
 8068  040776  063033                          ERR052                  ;  POSITIONING
 8069  041000  004037  031456                  JSR     R0,NERSTT       ;TAKE OTHER ERROR STATICS
 8070  041004  040770                          5$                      ;NO RECOVERY RETURN
 8071  041006  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
 8072  041014  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;ISSUE RECALIBRATE
 8073  041022  004037  051564                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 8074  041026  001324                          CINITQ                  ;  INITIATION QUEUE
 8075  041030  000207                          RTS     PC              ;RETURN
 8076
 8077  041032  032765  000020  000150  EC5$:   BIT     #BIT4,P.DSTT(R5) ;CHECK IF HEADER ERROR
 8078  041040  001002                          BNE     1$              ;YES, GO PROCESS ERROR
 8079  041042  000137  041556                  JMP     EC6$            ;NO, CONTINUE
 8080
 8081  041046  042765  000020  000150  1$:     BIC     #BIT4,P.DSTT(R5) ;CLEAR READ ALL HEADERS ISSUED
 8082  041054  010346                          MOV     R3,-(SP)        ;STORE R3 ON THE STACK
 8083  041056  016546  000056                  MOV     P.B11(R5),-(SP) ;STORE SECTOR READ
 8084  041062  042716  177017                  BIC     #^C<M.SECT>,(SP) ;KEEP ONLY SECTOR COUNT
 8085  041066  006216                          ASR     (SP)            ;SHIFT 4 BITS RIGHT
 8086  041070  006216                          ASR     (SP)
 8087  041072  006216                          ASR     (SP)
 8088  041074  006216                          ASR     (SP)            ;GET SECTOR COUNT
 8089  041076  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
 8090  041100  116516  000004                  MOVB    P.SECT(R5),(SP) ;GET DESIRED SECTOR
 8091  041104  162616                          SUB     (SP)+,(SP)      ;CALCULATE DIFFERENCE
 8092  041106  100410                          BMI     2$              ;BRANCH IF IN LOWER PART OF BUFFER
 8093  041110  012703  066602                  MOV     #BUFADD-6,R3    ;LOAD BASE OF HIGHER PART OF BUFFER
 8094  041114  006316                          ASL     (SP)            ;MULTIPLY SECTOR COUNT BY 6
 8095  041116  011646                          MOV     (SP),-(SP)
 8096  041120  006316                          ASL     (SP)
 8097  041122  062616                          ADD     (SP)+,(SP)
 8098  041124  162603                          SUB     (SP)+,R3        ;CALCULATE HEADER ADDRESS
 8099  041126  000407                          BR      3$              ;CHECK HEADER
 8100
 8101  041130  005416                  2$:     NEG     (SP)            ;MAKE COUNT POSITIVE
 8102  041132  006316                          ASL     (SP)            ;MULTIPLY DIFFERENCE BY 6
 8103  041134  011603                          MOV     (SP),R3
 8104  041136  006316                          ASL     (SP)
 8105  041140  062603                          ADD     (SP)+,R3
 8106  041142  062703  066376                  ADD     #HDBUFF-6,R3    ;CALCULATE HEADER ADDRESS
 8107  041146  012337  001646          3$:     MOV     (R3)+,HEAD1     ;GET FIRST WORD OF HEADER
 8108  041152  012337  001650                  MOV     (R3)+,HEAD2     ;GET SECOND WORD OF HEADER
 8109  041156  011337  001652                  MOV     (R3),HEAD3      ;GET THIRD WORD OF HEADER
 8110  041162  012603                          MOV     (SP)+,R3        ;RESTORE R3
 8111  041164  004737  037154                  JSR     PC,RSTAT        ;RESTORE STATUS
 8112  041170  013746  001646                  MOV     HEAD1,-(SP)     ;STORE FIRST WORD OF THE HEADER
 8113  041174  043716  001650                  BIC     HEAD2,(SP)      ;HD1 & ^HD2
 8114  041200  013746  001650                  MOV     HEAD2,-(SP)     ;STORE SECOND WORD OF HEADER
 8115  041204  043716  001646                  BIC     HEAD1,(SP)      ;^HD1 & HD2
 8116  041210  052616                          BIS     (SP)+,(SP)      ;GENERATE EXPECTED HEADER VRC
```

```
 8117  041212  023726  001652                    CMP     HEAD3,(SP)+     ;CHECK IF HEADER VRC IS CORRECT
 8118  041216  001501                            BEQ     15$             ;NO, CONTINUE
 8119  041220  004037  031032                    JSR     R0,PRI200       ;PRINT HEADER VRC ERROR
 8120  041224  062660                            ERR043
 8121  041226  105765  000120                    TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
 8122  041232  001016                            BNE     5$              ;YES, INDICATE UNSUCCESSFUL RECOVERY
 8123  041234  105265  000147            4$:     INCB    P.RERD(R5)      ;INCREMENT REREAD COUNT
 8124  041240  122765  000043  000147            CMPB    #43,P.RERD(R5)  ;CHECK IF UNRECOVERABLE
 8125  041246  101410                            BLOS    5$              ;YES, INDICATE UNRECOVERABLE ERROR
 8126  041250  122765  000121  000123            CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA COMMAND
 8127  041256  001407                            BEQ     6$              ;YES, START ERROR RECOVERY
 8128  041260  122765  000021  000147            CMPB    #21,P.RERD(R5)  ;CHECK IF UNRECOVERABLE WRITE
 8129  041266  101021                            BHI     8$              ;NO, START ERROR RECOVERY
 8130  041270  004737  031660            5$:     JSR     PC,UNRECY       ;INDICATE UNRECOVERABLE ERROR
 8131  041274  000207                            RTS     PC              ;RETURN
 8132
 8133  041276  122765  000021  000147    6$:     CMPB    #21,P.RERD(R5)  ;CHECK IF BEGINNING OFFSETTING OPERATION
 8134  041304  001007                            BNE     7$              ;NO, DETERMINE PROPER OFFSET
 8135  041306  010446                            MOV     R4,-(SP)        ;STORE R4 ON THE STACK
 8136  041310  116504  000000                    MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
 8137  041314  146437  001514  001513            BICB    INTMSK(R4),O.OVER ;DO NOT DO IMPLIED SEEKS
 8138  041322  012604                            MOV     (SP)+,R4        ;RESTORE R4
 8139  041324  004037  037204            7$:     JSR     R0,DETOFF       ;DETERMINE OFFSET VALUE
 8140  041330  041420                            12$                     ;OFFSET ISSUED RETURN
 8141  041332  116565  000123  000001    8$:     MOVB    P.RCMD(R5),P.CMND(R5) ;STORE COMMAND
 8142  041340  013765  001642  000002            MOV     RTYCYL,P.CYLN(R5) ;GET CYLINDER
 8143  041346  013765  001640  000004            MOV     RTYSEC,P.SECT(R5) ;GET TRACK AND SECTOR
 8144  041354  013765  001634  000010            MOV     RTYBA,P.BALO(R5) ;GET BUS ADDRESS
 8145  041362  013765  001636  000012            MOV     RTYWC,P.WC(R5) ;GET WORD COUNT
 8146  041370  005465  000012                    NEG     P.WC(R5)        ;MAKE IT POSITIVE
 8147  041374  122765  000131  000001            CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK COMMAND
 8148  041402  001003                            BNE     10$             ;NO, GO ISSUE COMMAND
 8149  041404  052765  000200  000014            BIS     #W.WCK,P.PRST(R5) ;ISSUE WRITE BEFORE WRITE CHECK
 8150  041412  004037  051564            10$:    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK ON COMMAND
 8151  041416  001324                            CINITQ                  ;  INITIATION QUEUE
 8152  041420  000207                    12$:    RTS     PC              ;RETURN
 8153
 8154  041422  023765  001646  000030    15$:    CMP     HEAD1,P.DCYL(R5) ;CHECK IF CYLINDER CORRECT
 8155  041430  001420                            BEQ     20$             ;YES, CHECK SECTOR TRACK AND FORMAT
 8156  041432  004037  031032                    JSR     R0,PRI200       ;PRINT MISPOSITIONING
 8157  041436  063473                            ERR064
 8158  041440  004037  031524                    JSR     R0,OPISTT       ;GO TAKE STATISTICS
 8159  041444  041470                            17$                     ;UNRECOVERABLE ERROR RETURN
 8160  041446  052765  004000  000150            BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
 8161  041454  112765  000113  000001            MOVB    #RECAL,P.CMND(R5) ;ISSUE RECALIBRATE
 8162  041462  004037  051564                    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
 8163  041466  001324                            CINITQ                  ;  COMMAND INITIATION QUEUE
 8164  041470  000207                    17$:    RTS     PC              ;RETURN
 8165
 8166  041472  005046                    20$:    CLR     -(SP)           ;MAKE ROOM ON STACK
 8167  041474  113766  001641  000001            MOVB    RTYTRK,1(SP)    ;GET TRACK
 8168  041502  006216                            ASR     (SP)            ;SHIFT RIGHT 3 BITS
 8169  041504  006216                            ASR     (SP)
 8170  041506  006216                            ASR     (SP)
 8171  041510  153716  001640                    BISB    RTYSEC,(SP)     ;GENERATE EXPECTED SECOND WORD
 8172  041514  052716  140000                    BIS     #140000,(SP)    ;PUT IN GO TRACK BITS
```

```
8173  041520  022637  001650                    CMP     (SP)+,HEAD2     ;CHECK IF SECOND WORD IS CORRECT
8174  041524  001407                            BEQ     25$             ;CHECK FOR BAD SECTOR
8175  041526  004037  031032                    JSR     R0,PRI200       ;INDICATE OPERATION INCOMPLETE
8176  041532  062641                            ERR042
8177  041534  004037  031524                    JSR     R0,OPISTT       ;GO TAKE STATISTICS
8178  041540  041470                            17$                     ;UNRECOVERABLE RETURN
8179  041542  000673                            BR      8$              ;GO ISSUE LAST COMMAND
8180
8181  041544  004037  031032          25$:      JSR     R0,PRI200       ;TYPE HEADER ERROR
8182  041550  064630                            ERR209
8183  041552  000137  041234                    JMP     4$              ;GO TRY TO RETRY COMMAND
8184
8185  041556  122765  000021  000147  EC6$:     CMPB    #21,P.RERD(R5)  ;CHECK IF REISSUE DATA TRANSFER
8186  041564  101036                            BHI     10$             ;YES, INDICATE SUCESSFUL RECOVERY
8187  041566  001424                            BEQ     5$              ;NO, OFFSET = +400 MICRO-INCHES
8188                                                                    ;   ISSUE DATA TRANSFER
8189  041570  122765  000024  000147            CMPB    #24,P.RERD(R5)  ;CHECK IF OFFSET -400 MICRO-INCHES
8190  041576  001420                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
8191  041600  122765  000027  000147            CMPB    #27,P.RERD(R5)  ;CHECK IF OFFSET +800 MICRO-INCHES
8192  041606  001414                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
8193  041610  122765  000032  000147            CMPB    #32,P.RERD(R5)  ;CHECK IF OFFSET -800 MICRO-INCHES
8194  041616  001410                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
8195  041620  122765  000035  000147            CMPB    #35,P.RERD(R5)  ;CHECK IF OFFSET +1200 MICRO-INCHES
8196  041626  001404                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
8197  041630  122765  000040  000147            CMPB    #40,P.RERD(R5)  ;CHECK IF OFFSET -1200 MICRO-INCHES
8198  041636  001011                            BNE     10$             ;NO, INDICATE SUCESSFUL RECOVERY
8199  041640  105265  000147          5$:       INCB    P.RERD(R5)      ;INCREMENT REREAD COUNT
8200  041644  116565  000123  000001            MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE COMMAND
8201  041652  004037  051564                    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
8202  041656  001324                            CINITQ                  ;   INITIATION QUEUE
8203  041660  000207                            RTS     PC              ;RETURN
8204
8205  041662  032765  040000  000150  10$:      BIT     #BIT14,P.DSTT(R5) ;CHECK IF SERVICING BAD SECTOR
8206  041670  001403                            BEQ     11$             ;NO, CONTINUE
8207  041672  005765  000146                    TST     P.RECT(R5)      ;CHECK IF IN RETRY SEQUENCE
8208  041676  001472                            BEQ     EC7$            ;NO, GO TO BAD SECTOR SERVICE ROUTINE
8209  041700  105765  000210          11$:      TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
8210  041704  001403                            BEQ     12$             ;NO, CHECK IF SEEK TO PREVIOUS CYLINDER
8211  041706  004737  036454                    JSR     PC,SURECY       ;INIDICATE SUCCESSFUL RECOVERY
8212  041712  000207                            RTS     PC              ;RETURN
8213
8214  041714  122765  000117  000001  12$:      CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK TO PREVIOUS CYLINDER
8215  041722  001404                            BEQ     14$             ;YES, INDICATE SUCCESSFUL RECOVERY
8216  041724  122765  000140  000001            CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE
8217  041732  001003                            BNE     16$             ;NO, PROCESS DATA TRANSFER
8218  041734  004737  036454          14$:      JSR     PC,SURECY       ;INDICATE SUCCESSFUL RECOVERY
8219  041740  000207                            RTS     PC              ;RETURN
8220
8221  041742  004037  025210          16$:      JSR     R0,CHKADD       ;CHECK SECTOR, TRACK, CYLINDER, BUS ADDRESS
8222                                                                    ;   AND WORD COUND
8223  041746  042004                            20$                     ;ERROR RETURN
8224  041750  005037  177776                    CLR     PS              ;ALLOW RK06 INTERRUPTS
8225  041754  122765  000121  000001            CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ DATA
8226  041762  001021                            BNE     30$             ;NO, CHECKJ IF ;WRITE CHECK
8227  041764  004037  023462                    JSR     R0,BUFER1       ;FIND FIRST DATA ERROR
8228  041770  042006                            25$                     ;ERROR RETURN
```

```
8229  041772  013737  001452  177776  18$:    MOV     RKPRI,PS          ;LOCK OUT RK06 INTERRUPTS
8230  042000  004737  036454                  JSR     PC,SURECY         ;INDICATE SUCCESSFUL RECOVERY
8231  042004  000207                  20$:    RTS     PC                ;RETURN
8232
8233  042006  013737  001452  177776  25$:    MOV     RKPRI,PS          ;LOCK OUT RK06 INTERRUPTS
8234  042014  052765  000004  000212          BIS     #BIT2,P.ERR(R5)   ;INDICATE DATA COMPARE ERROR
8235  042022  000137  035466                  JMP     E10$              ;PROCESS DATA COMPARE ERROR
8236
8237  042026  122765  000131  000001  30$:    CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
8238  042034  001356                          BNE     18$               ;NO, INDICATE SUCCESSFUL RECOVERY
8239  042036  032765  000200  000014          BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK ISSUED
8240  042044  001752                          BEQ     18$               ;YES, INDICATE SUCCESSFUL RECOVERY
8241  042046  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;CLEAR WRITE BEFORE WRITE CHECK
8242  042054  004037  051564                  JSR     R0,Q.PUSH         ;ENQUEUE COMMAND IN COMMAND INITIATION QUEUE
8243  042060  001324                          CINITQ
8244  042062  000207                          RTS     PC                ;RETURN
8245
8246  042064  004037  025210          EC7$:   JSR     R0,CHKADD         ;CHECK IF BUS ADD, WORD COUNT, SECTOR,
8247                                                                    ;  TRACK AND CYLINDER ARE CORRECT
8248  042070  042156                          9$                        ;ERROR RETURN
8249  042072  042765  040000  000150          BIC     #BIT14,P.DSTT(R5) ;CLEAR SERVICING BAD SECTOR
8250  042100  004737  042320                  JSR     PC,BSTRAN         ;CALCULATE NUMBER OF WORDS TRANSFERRED
8251  042104  013765  001660  000012          MOV     BSWC,P.WC(R5)     ;LOAD BUS ADDRESS FOR NEXT DATA TRANSFER
8252  042112  100022                          BPL     10$               ;CHECK IF DATA TRANSFER FINISHED
8253  042114  013765  001654  000004          MOV     BSSECT,P.SECT(R5) ;LOAD SECTOR AND TRACK FOR NEXT COMMAND
8254  042122  013765  001656  000002          MOV     BSCYLN,P.CYLN(R5) ;LOAD CYLINDER FOR NEXT COMMAND
8255  042130  013765  001662  000010          MOV     BSBA,P.BALO(R5)   ;LOAD BUS ADDRESS FOR NEXT DATA TRANSFER
8256  042136  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;GO ISSUE WRITE
8257  042144  004037  051564          5$:     JSR     R0,Q.PUSH         ;ENQUEUE COMMAND IN COMMAND INITIATION
8258  042150  001324                          CINITQ                    ;  QUEUE
8259  042152  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR PROCESSING FLAG
8260  042156  000207                  9$:     RTS     PC                ;RETURN
8261
8262  042160  105765  000210          10$:    TSTB    P.ASSN(R5)        ;CHECK IF PACK IS BEING WRITTEN
8263  042164  001012                          BNE     13$               ;YES, CHECK IF SEEK TO NEXT CYLINDER
8264  042166  004737  023020                  JSR     PC,BUFREL         ;RELEASE BUFFER
8265  042172  004037  051564                  JSR     R0,Q.PUSH         ;ENQUEUE PARMETER BLOCK IN DRIVE
8266  042176  001314                          AVAILQ                    ;  AVAILIABLE QUEUE
8267  042200  004737  022520                  JSR     PC,GETBUF         ;GET NEXT BUFFER FOR DAT TRANSFER
8268  042204  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR BEING PROCESSING
8269  042210  000207                          RTS     PC                ;RETURN
8270
8271  042212  005037  001566          13$:    CLR     ERRPRO            ;CLEAR ERROR BEING PROCESSED
8272  042216  013765  001654  000026          MOV     BSSECT,P.DTS(R5)  ;LOAD NEXT SECTOR AND TRACK
8273  042224  013765  001656  000030          MOV     BSCYLN,P.DCYL(R5) ;LOAD NEXT CYLINDER
8274  042232  032777  040000  136700          BIT     #SW14,@SWR        ;CHECK IF LOOP ON CURRENT OPERATION
8275  042240  001002                          BNE     14$               ;YES, ISSUE SEEK TO PREVIOUS CYLINDER
8276  042242  000137  015454                  JMP     WV0$              ;NO, CONTINUE TO WRITE PACK
8277
8278  042246  004737  023020          14$:    JSR     PC,BUFREL         ;RELEASE BUFFER
8279  042252  032765  002000  000014          BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
8280  042260  001402                          BEQ     15$               ;NO ISSUE SEEK TO PREVIOUS CYLINDER
8281  042262  000137  016216                  JMP     WV1$              ;DROP DRIVE
8282
8283  042266  112765  177777  000211  15$:    MOVB    #-1,P.SEEK(R5)    ;SET SEEK TO PREVIOUS CYLINDER
8284  042274  016565  000136  000002          MOV     P.LCYL(R5),P.CYLN(R5) ;GET LAST CYLINDER
```

```
 8285  042302  016565  000140  000004          MOV     P.LSEC(R5),P.SECT(R5) ;GET LAST TRACK AND SECTOR
 8286  042310  112765  000117  000001          MOVB    #SEEK,P.CMND(R5) ;LOAD COMMAND
 8287  042316  000712                          BR      5$              ;GO ISSUE SEEK COMMAND
```

```
                                        .SBTTL  CALCULATE WORDS TRANSFERRED UNTIL BAD SECTOR
                                        
042320  016537  000026  001654  BSTRAN: MOV     P.DTS(R5),BSSECT ;STORE ADDRESS OF BAD SECTOR
042326  016537  000030  001656          MOV     P.DCYL(R5),BSCYLN
042334  105237  001654                  INCB    BSSECT           ;ADVANCE TO NEXT SECTOR ON PACK
042340  122737  000025  001654          CMPB    #21.,BSSECT      ;CHECK FOR SECTOR OVERFLOW
042346  103014                          BHIS    5$               ;NO, CONTINUE
042350  105037  001654                  CLRB    BSSECT           ;CLEAR SECTOR
042354  105237  001655                  INCB    BSTRCK           ;INCREMENT TRACK
042360  122737  000002  001655          CMPB    #2,BSTRCK        ;CHECK FOR TRACK OVERFLOW
042366  103004                          BHIS    5$               ;NO, CONTINUE
042370  105037  001655                  CLRB    BSTRCK           ;CLEAR TRACK
042374  005237  001656                  INC     BSCYLN           ;INCREMENT CYLINDER
042400  013746  001656          5$:     MOV     BSCYLN,-(SP)     ;STORE CYLINDER ON STACK
042404  166516  000002                  SUB     P.CYLN(R5),(SP)  ;SUBTRACT STARTING CYLINDER
042410  011646                          MOV     (SP),-(SP)       ;SAVE DIFFERENCE
042412  006316                          ASL     (SP)             ;MULTIPLY CYLINDER BY 3
042414  062616                          ADD     (SP)+,(SP)
042416  005046                          CLR     -(SP)            ;MAKE ROOM ON STACK
042420  113716  001655                  MOVB    BSTRCK,(SP)      ;PLACE TRACK ON STACK
042424  062616                          ADD     (SP)+,(SP)       ;ADD TRACK
042426  005046                          CLR     -(SP)            ;MAKE ROOM ON THE STACK
042430  116516  000005                  MOVB    P.TRCK(R5),(SP)  ;STORE TRACK
042434  162616                          SUB     (SP)+,(SP)       ;SUBTRACT INITIAL TRACK
042436  012746  000026                  MOV     #22.,-(SP)       ;STORE 22 ON STACK FOR MULTIPLICATION
042442  004737  055210                  JSR     PC,$MULT         ;MULTIPLY BY 22
042446  005046                          CLR     -(SP)            ;MAKE ROOM ON THE STACK
042450  113716  001654                  MOVB    BSSECT,(SP)      ;STORE BAD SECTOR
042454  062616                          ADD     (SP)+,(SP)       ;ADD BAD SECTOR ADDRESS
042456  005046                          CLR     -(SP)            ;MAKE ROOM ON THE STACK
042460  116516  000004                  MOVB    P.SECT(R5),(SP)  ;STORE ORGINAL SECTOR ADDRESS
042464  162616                          SUB     (SP)+,(SP)       ;SUBTRACT ORGINAL SECTOR ADDRESS
042466  111666  000001                  MOVB    (SP),1(SP)       ;KEEP NUMBER OF SECTORS TRANSFERRED
042472  105016                          CLRB    @(SP)
042474  011637  001664                  MOV     (SP),BSWORD      ;LOAD WORD COUNT
042500  006316                          ASL     (SP)             ;MULTIPLY BY 2
042502  066516  000010                  ADD     P.BALO(R5),(SP)  ;GET STARTING ADDRESS FOR NEW DATA
042506  011637  001662                  MOV     (SP),BSBA        ;GET NEW BUS ADDRESS
042512  166516  000132                  SUB     P.RBAL(R5),(SP)  ;GET NUMBER OF BYTES TRANSFERRED
042516  006216                          ASR     (SP)             ;CONVERT TO WORDS (DIVIDE BY 2)
042520  066516  000130                  ADD     P.RWC(R5),(SP)   ;DETERMINE REMAINING WORD COUNT
042524  012637  001660                  MOV     (SP)+,BSWC       ;STORE CALCULATED WORD COUNT
042530  005726                          TST     (SP)+            ;THROW AWAY MOST SIGNIFICANT BITS
042532  000207                          RTS     PC               ;RETURN
```

```
                                        .SBTTL  BAD SECTOR HANDLING
                                
042534  032765  040000  000150  BDSECT: BIT     #BIT14,P.DSTT(R5) ;CHECK IF PRESENTLY SERVICING BAD SECTOR
042542  001401                          BEQ     5$                ;NO, CONTINUE
042544  000000                          HALT
042546  132765  000010  000210  5$:     BITB    #BIT3,P.ASSN(R5) ;CHECK IF READING PACK SERIAL NUMBER
042554  001157                          BNE     40$               ;YES, DETERMINE NEXT SECTOR
042556  004737  042320                  JSR     PC,BSTRAN         ;CALCULATE WORDS TRANSFERRED
042562  122765  000121  000001          CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ COMMAND
042570  001076                          BNE     25$               ;NO, DO NO CHECK BUS ADDRESS AND WORD COUNT
042572  013746  001662                  MOV     BSBA,-(SP)        ;STORE BUS ADDRESS
042576  162716  001000                  SUB     #256.*2,(SP)      ;SUBTRACT ADDITIONAL SECTOR
042602  026526  000024                  CMP     P.BAR(R5),(SP)+   ;CHECK IF BUS ADDRESS CORRECT
042606  001404                          BEQ     10$               ;YES, CONTINUE
042610  004037  037500                  JSR     R0,CONERR         ;PRINT 'BUS ADDRESS INVALID''
042614  063612                          ERR071
042616  000207                          RTS     PC                ;RETURN
                                
042620  016546  000022          10$:    MOV     P.WCR(R5),-(SP) ;STORE WORD COUNT FOR COMPARISON
042624  062716  000400                  ADD     #256.,(SP)        ;SUBTRACT OFF BAD SECTOR
042630  023726  001660                  CMP     BSWC,(SP)+        ;CHECK IF WORD COUNT CORRECT
042634  001404                          BEQ     15$               ;YES, CONTINUE
042636  004037  037500                  JSR     R0,CONERR         ;PRINT 'WORD COUNT INVALID''
042642  063567                          ERR070
042644  000207                          RTS     PC                ;RETURN
                                
042646  042765  100200  000150  15$:    BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
042654  022737  000400  001664          CMP     #256.,BSWORD      ;CHECK IF NO WORDS TRANSFERRED
042662  001413                          BEQ     20$               ;YES, DO NOT CHECK DATA
042664  004037  023462                  JSR     R0,BUFER1         ;CHECK DATA READ
042670  043224                          50$                       ;ERROR RETURN
042672  063765  001664  000012          ADD     BSWORD,P.WC(R5) ;CALCULATE NEW WORD COUNT
042700  100410                          BMI     22$               ;CHECK IF DATA TRANSFER NOT COMPLETE
042702  000464                          BR      30$               ;DATA TRANSFER COMPLETE
                                
042704  052765  000200  000014  19$:    BIS     #W.WCK,P.PRST(R5) ;ISSUE WRITE BEFORE WRITE CHECK
042712  013765  001660  000012  20$:    MOV     BSWC,P.WC(R5)     ;GET NUMBER OF WORDS TO BE TRANSFERRED
042720  100055                          BPL     30$               ;CHECK IF DATA TRANSFER COMPLETE
042722  013765  001656  000002  22$:    MOV     BSCYLN,P.CYLN(R5) ;GET NEW CYLINDER
042730  013765  001654  000004          MOV     BSSECT,P.SECT(R5) ;GET NEW SECTOR
042736  013765  001662  000010          MOV     BSBA,P.BALO(R5) ;GET NEW BUS ADDRESS
042744  004037  051564                  JSR     R0,Q.PUSH         ;PUT PARAMETER IN COMMAND
042750  001324                          CINITQ                    ;  INITIATION QUEUE
042752  005765  000146                  TST     P.RECT(R5)        ;CHECK IF ERROR RECOVERY IN PROGRESS
042756  001002                          BNE     23$               ;YES, DO NOT RESET ERROR PROCESSING IN PROGRESS
042760  005037  001566                  CLR     ERRPRO            ;RESET ERROR PROCESSING IN PROGRESS
042764  000207                  23$:    RTS     PC                ;RETURN
                                
042766  042765  100200  000150  25$:    BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
042774  122765  000013  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK COMMAND
043002  001343                          BNE     20$               ;NO, CONTINUE
043004  022737  000400  001664          CMP     #256.,BSWORD      ;CHECK IF NO WORDS TRANSFERRED
043012  001734                          BEQ     19$               ;YES, ISSUE NEXT DATA TRANSFER
043014  012746  000400                  MOV     #256.,-(SP)       ;STORE EXCESS WORD COUNT ON STACK
043020  163716  001664                  SUB     BSWORD,(SP)       ;CALCULATE NEW WORD COUNT
043024  012665  000012                  MOV     (SP)+,P.WC(R5)    ;STORE NEW WORD COUNT
```

```
 8388  043030  052765  040000  000150          BIS     #BIT14,P.DSTT(R5) ;INDICATE THAT BAD
 8389                                                                    ;  SECTOR IS BEING PROCESSED
 8390  043036  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;DO WRITE CHECK PART OF COMMAND
 8391  043044  004037  051564                  JSR     R0,Q.PUSH         ;PUT PARAMETER BLOCK IN COMMAND
 8392  043050  001324                          CINITQ                    ;  INITIATION QUEUE
 8393  043052  000207                          RTS     PC                ;RETURN
 8394
 8395  043054  004737  023020          30$:    JSR     PC,BUFREL         ;RELEASE BUFFER
 8396  043060  005765  000146                  TST     P.RECT(R5)        ;DETERMINE IF CURRENTLY UNDERGOING ERROR
 8397                                                                    ;  RECOVERY
 8398  043064  001403                          BEQ     35$               ;YES, PUT IN AVAILIABLE QUEUE
 8399  043066  004737  036454                  JSR     PC,SURECY         ;INDICATE SUCCESSFUL RECOVERY
 8400  043072  000207                          RTS     PC                ;RETURN
 8401
 8402  043074  004037  051564          35$:    JSR     R0,Q.PUSH         ;ENQUEUE PARAMETER BLOCK IN
 8403  043100  001314                          AVAILQ                    ;  AVAILIABLE QUEUE
 8404  043102  004737  022520                  JSR     PC,GETBUF         ;GET NEXT BUFFER
 8405  043106  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR BEING PROCESSED
 8406  043112  000207                          RTS     PC                ;RETURN
 8407
 8408  043114  042765  100200  000150  40$:    BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
 8409  043122  122737  000010  000126          CMPB    #8.,P.RSEC        ;CHECK IF ALL SECTORS HAVE BEEN READ
 8410  043130  101422                          BLOS    45$               ;YES, DROP DRIVE FROM TEST SEQUENCE
 8411  043132  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
 8412  043140  016565  000124  000136          MOV     P.RCYL(R5),P.LCYL(R5)
 8413  043146  016565  000126  000140          MOV     P.RSEC(R5),P.LSEC(R5)
 8414  043154  016537  000130  000142          MOV     P.RWC(R5),P.L'C
 8415  043162  004037  051564                  JSR     R0,Q.PUSH         ;ENQUEUE PARAMETER BLOCK IN COMMAND
 8416  043166  001324                          CINITQ                    ;  INITIATION QUEUE
 8417  043170  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR RECOVERY IN PROGRESS
 8418  043174  000207                          RTS     PC                ;RETURN
 8419
 8420  043176  004737  023020          45$:    JSR     PC,BUFREL         ;GO RELEASE BUFFER
 8421  043202  104401  064410                  TYPE    ,ERR204
 8422  043206  004737  013162                  JSR     PC,DROP           ;DROP DRIVE FROM TEST SEQUENCE
 8423  043212  004737  022520                  JSR     PC,GETBUF         ;GET NEW BUFFER
 8424  043216  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR RECOVERY IN PROGRESS
 8425  043222  000207                          RTS     PC                ;RETURN
 8426
 8427  043224  052765  000004  000212  50$:    BIS     #BIT2,P.ERR(R5)   ;SET DATA TYPE ERROR
 8428  043232  000137  035466                  JMP     E10$              ;GO PROCESS DATA ERROR
```

```
                                        .SBTTL  ERROR RECOVERY SEQUENCE ABNORMAL RETURN

043236  032765  000120  000150  ERCVY2: BIT     #BIT4!BIT5,P.DSTT(R5) ;CHECK IF DIAGNOSING SEQUENCE
043244  001524                          BEQ     35$             ;NO, GO TO ERROR RECOVERY SEQUENCE
043246  004737  030156                  JSR     PC,PRI000       ;PRINT HEADER
043252  104401  064577                  TYPE    ,ERR208         ;TYPE 'ERROR WHILE DIAGNOSING ---''
043256  032765  000020  000150          BIT     #BIT4,P.DSTT(R5) ;CHECK IF HEADER TYPE ERROR
043264  001005                          BNE     2$              ;YES, PRINT MESSAGE
043266  104401  063034                  TYPE    ,ERR052+1       ;TYPE 'TIME OUT WHILE DRIVE POSITION''
043272  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
043276  000402                          BR      3$              ;GO PRINT STATUS

043300  104401  064631          2$:     TYPE    ,ERR209+1       ;PRINT HEADER TYPE ERROR
043304  032777  020000  135626  3$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
043312  001066                          BNE     30$             ;YES, DO NOT PRINT STATUS
043314  032765  000020  000212          BIT     #BIT4,P.ERR(R5) ;CHECK IF DRIVE SEIZED TIME OUT
043322  001402                          BEQ     4$
043324  104401  063351                  TYPE    ,ERR059
043330  032765  000040  000212  4$:     BIT     #BIT5,P.ERR(R5) ;CHECK IF TIME OUT WHILE DRIVE POSITIONING
043336  001402                          BEQ     5$
043340  104401  063034                  TYPE    ,ERR052+1
043344  032765  000010  000014  5$:     BIT     #UEXATT,P.PRST(R5) ;CHECK IF UNEXPECTED ATTENTION
043352  001402                          BEQ     6$
043354  104401  062430                  TYPE    ,ERR030
043360  032765  000020  000014  6$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF DRIVE HARD ERROR
043366  001402                          BEQ     7$
043370  104401  062321                  TYPE    ,ERR027
043374  032765  000040  000014  7$:     BIT     #DRVDSC,P.PRST(R5) ;CHECK IF DSC DID NOT CLEAR
043402  001402                          BEQ     8$
043404  104401  062340                  TYPE    ,ERR028
043410  032765  004000  000014  8$:     BIT     #NODSC,P.PRST(R5) ;CHECK IF ATTN BUT NO DSC OR FAULT
043416  001402                          BEQ     11$
043420  104401  063274                  TYPE    ,ERR058
043424  004737  026252          11$:    JSR     PC,PRISTT       ;PRINT CONTROLLER STATUS
043430  004737  026076                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
043434  004737  030422                  JSR     PC,PRI002       ;PRINT RETRY COMMAND
043440  010546                          MOV     R5,-(SP)        ;STORE R5 ON STACK
043442  012705  004516                  MOV     #PARM10,R5      ;LOAD R5 FOR PREVIOUS STATUS
043446  104401  064650                  TYPE    ,ERR210         ;TYPE 'FIRST ERROR INFO''
043452  004737  026252                  JSR     PC,PRISTT       ;PRINT CONTROLLER STATUS
043456  004737  026076                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
043462  004737  030422                  JSR     PC,PRI002       ;PRINT INITIAL COMMAND
043466  012605                          MOV     (SP)+,R5        ;RESTORE R5
043470  005265  000206          30$:    INC     P.NER(R5)       ;INCREMENT OTHER ERRORS
043474  004737  031426                  JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
043500  004737  023020                  JSR     PC,BUFREL       ;RELEASE BUFFER
043504  004737  013162                  JSR     PC,DROF         ;DROP DRIVE
043510  004737  022520                  JSR     PC,GETBUF       ;GO ALLOCATE BUFFERS
043514  000405                          BR      40$             ;CHECK IF ERROR QUEUE EMPTY

043516  004737  032160          35$:    JSR     PC,ERCVY        ;GO DO ERROR RECOVERY
043522  005737  001566                  TST     ERRPRO          ;CHECK IF ERROR RECOVERY ON LAST DRIVE
                                                                ;  IS FINISHED
043526  001007                          BNE     45$             ;NO, RETURN
043530  004037  051644          40$:    JSR     R0,Q.POP        ;GET NEXT ELEMENT FROM
043534  001330                          ERRPRQ                  ;  ERROR PROCESSING QUEUE
```

```
 8485  043536  012605                        MOV     (SP)+,R5        ;LOAD PARAMETER BLOCK
 8486  043540  010537  001566                MOV     R5,ERRPRO       ;LOAD CURRENT PROCESSING ERROR
 8487  043544  001364                        BNE     35$             ;IF ERROR PROCESSING QUEUE NOT EMPTY GO
 8488                                                                ;  PROCESS NEXT ERROR
 8489  043546  000207              45$:      RTS     PC              ;RETURN
```

```
                                        .SBTTL  DETERMINE IF CLOCK IS PRESENT

043550  012737  043600  000004  CHKCLK: MOV     #5$,ERRVEC      ;SET UP ERROR VECTOR FOR NON-EXISTENT MEMORY
043556  012737  000340  000006          MOV     #PR7,ERRVEC+2   ;LOCK ALL INTERRUPTS
043564  005777  135546                  TST     @$LKCSB         ;CHECK FOR P CLOCK
043570  112737  177777  001374          MOVB    #-1,CLKFLG      ;INDICATE KW11-P PRESENT
043576  000413                          BR      20$             ;LOAD COMMON CLOCK PARAMETERS

043600  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
043602  012737  043624  000004          MOV     #15$,ERRVEC     ;SET UP ERROR VECTOR FOR NON-EXISTENT MEMORY
043610  005777  135526                  TST     @$LKS           ;CHECK FOR L CLOCK
043614  112737  000001  001374          MOVB    #1,CLKFLG       ;SET KW11-L PRESENT
```

```
 8502  043622  000401                          BR      20$             ;REINSTATE TRAP CATCHER
 8503
 8504  043624  022626                  15$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
 8505  043626  012737  000006  000004  20$:    MOV     #ERRVEC+2,ERRVEC ;REINSTATE TRAP CATCHER
 8506  043634  005037  000006                  CLR     ERRVEC+2
 8507  043640  000207                          RTS     PC              ;RETURN
```

```
 8508                                   .SBTTL  SET UP CLOCK INTERRUPT
 8509
 8510  043642  105737  JC1374          CLKINT: TSTB    CLKFLG          ;CHECK IF A CLOCK IS ON SYSTEM
 8511  043646  001431                          BEQ     20$             ;NO, RETURN
 8512  043650  100011                          BPL     5$              ;CHECK IF KW11-L
 8513  043652  013701  001340                  MOV     $LPVEC,R1       ;LOAD VECTOR ADDRESS
 8514  043656  012777  177777  135452          MOV     #-1,@$LKCSB     ;LOAD COUNT BUFFER WITH 1'S
 8515  043664  012777  000135  135442          MOV     #135,@$LKCSR    ;SET COUNT UP, 16 MS, CONT
 8516  043672  000405                          BR      10$             ;LOAD COMMON CLOCK PARAMETERS
 8517
 8518  043674  013701  001344          5$:     MOV     $LLVEC,R1       ;LOAD VECTOR ADDRESS
 8519  043700  012777  000100  135434          MOV     #100,@$LKS      ;SET UP L CLOCK FOR INTERRUPT MODE
 8520  043706  012721  043734          10$:    MOV     #CLOCK,(R1)+    ;LOAD ADDRESS OF INTERRUPT ROUTINE
 8521  043712  012711  000300                  MOV     #PR6,(R1)       ;SET UP FOR PRIORITY 6
 8522  043716  113737  001352  001353          MOVB    HZ,CLKFRQ       ;SET UP CLOCK FREQUENCY
 8523  043724  113737  001360  001375          MOVB    PERINV,PERIOD   ;SET UP COUNTER FOR INITIAL STATISTIC SUMMARY
 8524  043732  000207                  20$:    RTS     PC              ;RETURN
```

```
 8525                                   .SBTTL  KW11-L AND KW11-P INTERRUPT HANDLER
 8526
 8527  043734  105337  001353          CLOCK:  DECB    CLKFRQ          ;CHECK IF ONE SECOND PASTED
 8528  043740  001041                          BNE     10$             ;NO, RETURN
 8529  043742  113737  001352  001353          MOVB    HZ,CLKFRQ       ;REINITIALIZE SYSTEM CLOCK
 8530  043750  005237  001372                  INC     SECOND          ;INCREMENT SECOND COUNT
 8531  043754  022737  000074  001372          CMP     #60.,SECOND     ;CHECK IF MINUTE HAS OCCURRED
 8532  043762  001030                          BNE     10$             ;NO, RETURN
 8533  043764  005037  001372                  CLR     SECOND          ;CLEAR SECOND
 8534  043770  105737  001360                  TSTB    PERINV          ;CHECK IF INTERVAL STATISTICS
 8535  043774  001411                          BEQ     5$              ;NO, UPDATE MINUTE COUNT
 8536  043776  105337  001375                  DECB    PERIOD          ;DECREMENT PERIOD COUNT
 8537  044002  001006                          BNE     5$              ;CHECK IF END OF PERIOD, IF NOT UPDATE MINUTE COUNT
 8538  044004  113737  001360  001375          MOVB    PERINV,PERIOD   ;INITIALIZE PERIOD
 8539  044012  112737  177777  001633          MOVB    #-1,STATIS      ;INDICATE TIME FOR STATISTIC TYPE OUT
 8540  044020  005237  001370          5$:     INC     MINUTE          ;INCREMENT MINUTE COUNTER
 8541  044024  022737  000074  001370          CMP     #60.,MINUTE     ;CHECK IF HOUR HAS OCCURED
 8542  044032  001004                          BNE     10$             ;NO, RETURN
 8543  044034  005037  001370                  CLR     MINUTE          ;CLEAR MINUTE COUNT
 8544  044040  005237  001366                  INC     HOUR            ;INCREMENT HOUR COUNT
 8545  044044  000002                  10$:    RTI                     ;RETURN
```

```
                                        .SBTTL  PRINT TIME ROUTINE
                                
044046  105737  001374          PRITIM: TSTB    CLKFLG          ;CHECK IF CLOCK ON SYSTEM
044052  001455                          BEQ     50$             ;NO, RETURN
044054  010046                          MOV     R0,-(SP)        ;SAVE R0
044056  013746  177776                  MOV     PS,-(SP)        ;SAVE PSW
044062  012737  000340  177776          MOV     #PR7,PS         ;LOCK OUT CLOCK
044070  013737  001366  001376          MOV     HOUR,TIMHR      ;SAVE TIME FOR PRINT OUT
044076  013737  001370  001400          MOV     MINUTE,TIMMIN
044104  013737  001372  001402          MOV     SECOND,TIMSEC
044112  012637  177776                  MOV     (SP)+,PS        ;RESTORE PSW
044116  012700  065341                  MOV     #TIM001,R0      ;LOAD R0 FOR INDEX
044122  105010                          CLRB    (R0)            ;DETERMINE HUNDREDS OF HOURS
044124  013746  001376                  MOV     TIMHR,-(SP)     ;STORE HOURS FOR CONVERSION
044130  162716  000144          1$:     SUB     #100.,(SP)      ;CONVERT HUNDREDS
044134  100402                          BMI     2$              ;CHECK IF TOO SMALL
044136  105210                          INCB    (R0)
044140  000773                          BR      1$
                                
044142  062716  000144          2$:     ADD     #100.,(SP)
044146  152720  000060                  BISB    #60.,(R0)+      ;MAKE DIGIT ASCII
044152  004737  044210                  JSR     PC,CONHUN
044156  013716  001400                  MOV     TIMMIN,(SP)     ;LOAD STACK FOR MINUTES
044162  004737  044210                  JSR     PC,CONHUN
044166  013716  001402                  MOV     TIMSEC,(SP)     ;LOAD STACK FOR SECONDS
044172  004737  044210                  JSR     PC,CONHUN
044176  005726                          TST     (SP)+           ;ADJUST STACK
044200  104401  065334                  TYPE    ,TIM000         ;TYPE TIME:
044204  012600                          MOV     (SP)+,R0        ;RESTORE R0
044206  000207          50$:    RTS     PC                      ;RETURN
                                
                                        .SBTTL  CONVERT DECIMAL  LESS THAN 100
                                
044210  105010          CONHUN: CLRB    (R0)            ;ZERO DIGIT
044212  162766  000012  000002  1$:     SUB     #10.,2(SP)      ;CONVERT TENS
044220  100402                          BMI     5$
044222  105210                          INCB    (R0)
044224  000772                          BR      1$
                                
044226  062766  000012  000002  5$:     ADD     #10.,2(SP)
044234  152720  000060                  BISB    #60,(R0)+       ;MAKE ASCII
044240  105010                          CLRB    (R0)            ;ZERO DIGIT
044242  005766  000002                  TST     2(SP)
044246  001404                          BEQ     7$
044250  105210                  6$:     INCB    (R0)            ;CONVERT ONES
044252  005366  000002                  DEC     2(SP)
044256  001374                          BNE     6$
044260  152720  000060          7$:     BISB    #60,(R0)+       ;MAKE ASCII
044264  105720                          TSTB    (R0)+           ;SKIP OVER COLON
044266  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.
                                ;*
                                ;;*****************************************************************

 044270  010546                 W.WTCH: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
 044272  010446                         MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
 044274  010346                         MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
 044276  010246                         MOV     R2,-(SP)        ;SAVE R2 ON STACK
 044300  013746  177776                 MOV     PS,-(SP)        ;SAVE PROGRAM STATUS WORD ON STACK
 044304  005337  001466                 DEC     W.MTIM          ;DECREMENT MILLISECOND TIMER
 044310  001046                         BNE     20$             ;IF NOT ZERO RETURN
 044312  013737  001470  001466         MOV     W.MILI,W.MTIM   ;REINITIALIZE MILLISECOND TIMER
 044320  013737  001452  177776         MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 044326  013702  001446                 MOV     RKBAS,R2        ;LOAD BASE OF RK06 REGISTERS
 044332  005004                         CLR     R4              ;CLEAR INDEX OF DRIVE MASKS
 044334  005003                         CLR     R3              ;CLEAR INDEX OF DRIVE TIMES
                                                                 ;  AND PARAMETER BLOCK ADDRESSES
 044336  136437  001514  001512 1$:     BITB    INTMSK(R4),W.TIME ;CHECK IF DRIVE IS BEING TIMED
```

```
8652  044344  001422                        BEQ     6$              ;NO, GO TO NEXT DRIVE
8653  044346  005363  001544                DEC     W.DRV(R3)       ;DECREMENT DRIVE COUNT
8654  044352  001017                        BNE     6$              ;IF NOT TIME OUT, GO TO
8655                                                                ;  NEXT DRIVE
8656  044354  016305  001524                MOV     PBLKT(R3),R5    ;LOAD ADDRESS OF PARAMETER
8657                                                                ;  BLOCK TABLE FOR INDEXING
8658  044360  146437  001514  001512        BICB    INTMSK(R4),W.TIME ;RESET TIMING INDICATOR FOR DRIVE
8659  044366  052765  000100  000014        BIS     #CMDTO,P.PRST(R5) ;SET COMMAND TIME OUT
8660  044374  020537  001464                CMP     R5,O.WAIT       ;CHECK IF DRIVER IS WAITING FOR
8661                                                                ;  COMMAND COMPLETION
8662  044400  001002                        BNE     5$              ;NO, DO NOT ALTER WAITING FOR
8663                                                                ;  COMMAND COMPLETION
8664  044402  005037  001464                CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
8665  044406  004777  135044        5$:     JSR     PC,@A.ABNL      ;BRANCH TO ERROR ROUTINE
8666  044412  062703  000002        6$:     ADD     #2,R3           ;ADD 2 TO INDEX OF DRIVE TIMES
8667                                                                ;  AND PARAMETER BLOCK ADDRESSES
8668  044416  005204                        INC     R4              ;INCREMENT INDEX OF DRIVE MASK
8669  044420  022704  000010                CMP     #10,R4          ;CHECK IF ALL DRIVES ARE TESTED
8670  044424  001344                        BNE     1$              ;NO, CHECK NEXT DRIVE
8671  044426  012637  177776        20$:    MOV     (SP)+,PS        ;RESTORE PSW
8672  044432  012602                        MOV     (SP)+,R2        ;RESTORE R2
8673  044434  012603                        MOV     (SP)+,R3        ;RESTORE R3
8674  044436  012604                        MOV     (SP)+,R4        ;RESTORE R4
8675  044440  012605                        MOV     (SP)+,R5        ;RESTORE R5
8676  044442  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 WILL BE ISSUED
                                ;*          FOR THE QUEUED RK06 DRIVER.
                                ;*
                                ;*      THREE LINKS ARE PROVIDED TO THE DRIVING PROGRAM.
                                ;*      THEY ARE:
                                ;*
                                ;*      1.) A.NORM ADDRESS OF NORMAL RETURN (SUCESSFUL COMPLETION OF COMMAND)
                                ;*      2.) A.ABNL ADDRESS OF ABNORMAL RETURN (UNSUCESSFUL COMPLETION OF COMMAND)
                                ;*      3.) A.CONT ADDRESS OF CONTROL ERROR RETURN
                                ;*
                                ;*      FOR NORMAL AND ABNORMAL RETURNS, THE ADDRESS OF THE APPROPRIATE
                                ;*      PARAMETER BLOCK WILL BE IN R5.
                                ;*
                                ;*      FOR THE CONTROLLER ERROR RETURN, THE LOCATION E.CONT CONTAINS
                                ;*      THE REASON FOR THE CONTROLLER ERROR.
                                ;*
                                ;*      ROUTINES USED:
                                ;*              C.OPT   (QUEUED ONLY)
                                ;*              Q.PUSH (QUEUED ONLY)
                                ;*              Q.RMOV (QUEUED ONLY)
                                ;*              R.CONT (SEQUENTIAL ONLY)
                                ;*              R.NORM (SEQUENTIAL ONLY)
                                ;*              R.ABNL (SEQUENTIAL ONLY)
                                ;*              I.CSTS
                                ;*              I.STAT
                                ;*              I.ISSU
                                ;*              I.CCLR
                                ;*
                                ;;*****************************************************************

 044444  010546                 I.INTR: MOV     R5,-(SP)        ;STORE R5 ON THE STACK
 044446  010446                         MOV     R4,-(SP)        ;STORE R4 ON THE STACK
 044450  010346                         MOV     R3,-(SP)        ;STORE R3 ON THE STACK
 044452  010246                         MOV     R2,-(SP)        ;STORE R2 ON THE STACK
 044454  010146                         MOV     R1,-(SP)        ;STORE R1 ON THE STACK
 044456  010046                         MOV     R0,-(SP)        ;STORE R0 ON THE STACK
 044460  013702  001446                 MOV     RKBAS,R2        ;LOAD R2 TO ADDRESS RK06 REGISTER
 044464  016237  000010  001412         MOV     RKCS2(R2),T.CS2 ;STORE CS2
 044472  032737  001000  001412         BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
 044500  001407                         BEQ     1$              ;NO, CONTINUE PROCESSING
 044502  052737  100000  001462         BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT
 044510  004777  134744                 JSR     PC,@A.CONT      ;REPORT ERROR
```

```
 8733  044514  000137  047052                 JMP     I.RTRN          ;RETURN
 8734
 8735  044520  105737  001506          1$:    TSTB    I.ISRL          ;CHECK IF INTERRUPT OR RELEASE
 8736  044524  001410                         BEQ     6$              ;NO, CHECK IF DRIVE AVAILABLE
 8737  044526  100403                         BMI     5$              ;CHECK IF RELEASE COMMAND
 8738  044530  105037  001506                 CLRB    I.ISRL          ;YES,CLEAR FLAG
 8739  044534  000425                         BR      I.IOO           ;CONTINUE PROCESSING INTERRUPT
 8740
 8741  044536  105037  001506          5$:    CLRB    I.ISRL          ;CLEAR FLAG
 8742  044542  000137  045576                 JMP     I.ATTN          ;GO PROCESS DRIVE ATTENTIONS
 8743
 8744  044546  032737  010400  001412  6$:    BIT     #NED!UFE,T.CS2  ;CHECK FOR NON-EXISTENT DRIVE OR
 8745                                                                 ;  UNIT FIELD ERROR
 8746  044554  001415                         BEQ     7$              ;NO, WAIT FOR DUAL ACCESS INTERRUPT
 8747  044556  013704  001412                 MOV     T.CS2,R4        ;LOAD R4 FOR DRIVE NUMBER
 8748  044562  042704  177770                 BIC     #^C<DRVMSK>,R4  ;KEEP DRIVE BITS
 8749  044566  010403                         MOV     R4,R3           ;STORE DRIVE NUMBER FOR INDEX
 8750  044570  006303                         ASL     R3              ;MULTIPLY BY 2
 8751  044572  016305  001524                 MOV     PBLKT(R3),R5    ;STORE PARAMETER BLOCK ADDRESS
 8752  044576  016237  000000  001410         MOV     RKCS1(R2),T.CS1 ;LOAD TEMPORARY CS1 FOR STATUS REPORT
 8753  044604  000137  044760                 JMP     I.ERRC          ;REPORT ERROR
 8754
 8755  044610                          7$:
 8756
 8757  044610  013705  001464          I.IOO: MOV     O.WAIT,R5       ;LOAD PARAMETER BLOCK ADDRESS INTO R5
 8758  044614  001002                         BNE     2$              ;IS COMMAND WAITING PROCESSING
 8759                                                                 ;  YES, DO PROCESSING
 8760  044616  000137  045576                 JMP     I.ATTN          ;NO, PROCESS ATTENTION
 8761
 8762  044622  013704  001412          2$:    MOV     T.CS2,R4        ;STORE RKCS2 FOR DRIVE NUMBER
 8763  044626  042704  177770                 BIC     #^C<DPVMSK>,R4  ;MASK OUT UNNECESSARY BITS
 8764
 8765
 8766  044632  010403                         MOV     R4,R3           ;STORE DRIVE NUMBER FOR INDEX
 8767  044634  006303                         ASL     R3              ;MULTIPLY BY 2 FOR INDEX
 8768
 8769  044636  126504  000000                 CMPB    P.DRVN(R5),R4   ;CHECK IF DRIVE NUMBER IS EXPECTED
 8770  044642  001401                         BEQ     3$              ;YES, CONTINUE
 8771  044644  000000                         HALT                    ;NO, DRIVER ERROR
 8772  044646  122765  000164  000001  3$:    CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
 8773  044654  001002                         BNE     10$             ;NO, EXECUTE NORMAL DATA TRANSFER
 8774  044656  000137  045236                 JMP     I.HDAL          ;GO EXECUTE SPECIAL HEADER SEQUENCE
 8775
 8776  044662  005037  001464          10$:   CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
 8777  044666  005063  001544                 CLR     W.DRV(R3)       ;CLEAR WATCH-DOG TIME
 8778  044672  146437  001514  001512         BICB    INTMSK(R4),W.TIME ;RESET TIMING ON THIS DRIVE
 8779  044700  016237  000000  001410         MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
 8780  044706  032737  100000  001410         BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 8781  044714  001021                         BNE     I.ERRC          ;YES, PROCESS ERROR
 8782  044716  016237  000016  001424         MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 8783  044724  136437  001514  001425         BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF DRIVE ATTENTION SET
 8784  044732  001004                         BNE     15$             ;YES, REPORT ERROR
 8785  044734  004777  134514                 JSR     PC,@A.NORM      ;INDICATE NORMAL RETURN
 8786  044740  000137  047052                 JMP     I.RTRN          ;RESTORE REGISTERS
 8787
 8788  044744  052765  000010  000014  15$:    BIS     #UEXATT,P.PRST(R5) ;SET UNEXPECTED ATTENTION
```

```
 8789
 8790  044752  004737  047542          I.ERRA: JSR     PC,I.CSTS       ;STORE CONTROLLER STATUS
 8791  044756  000405                          BR      I.ERR           ;STORE PATTERN AND POSITION INFORMATION
 8792
 8793  044760  013765  001410  000016  I.ERRC: MOV     T.CS1,P.CS1(R5) ;GET ERROR RKCS1
 8794  044766  004737  047564                  JSR     PC,I.CST1       ;GET REST OF CONTROLLER STATUS
 8795  044772  016265  000032  000062  I.ERR:  MOV     RKECPT(R2),P.EPAT(R5) ;STORE ECC PATTERN
 8796  045000  016265  000030  000060          MOV     RKECPS(R2),P.EPOS(R5) ;STORE ECC POSITION
 8797  045006  004037  047070                  JSR     R0,I.CCLR       ;CLEAR CONTROLLER
 8798  045012  047052                          I.RTRN                  ;ERROR RETURN
 8799  045014  032765  010400  000020          BIT     #NED!UFE,P.CS2(R5) ;CHECK IF IT WAS NON-EXISTENT DRIVE OR
 8800                                                                  ;  UNIT FIELD ERROR
 8801  045022  001056                          BNE     5$              ;YES, REPORT ERROR
 8802  045024  032765  000001  000036          BIT     #DRA,P.DS(R5)   ;SEE IF DRIVE IS AVAIL
 8803  045032  001004                          BNE     4$              ;BR IF YES
 8804  045034  052765  010000  000014          BIS     #DRVSZD,P.PRST(R5) ;SET 'DRIVE SIEZED'' BIT
 8805  045042  000471                          BR      10$             ;REPORT ERROR
 8806  045044  004037  047646          4$:     JSR     R0,I.STAT       ;GATHER DRIVE STATUS
 8807  045050  047052                          I.RTRN                  ;ERROR RETURN
 8808  045052  112737  000005  001410          MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
 8809  045060  004037  047152                  JSR     R0,I.ISSU       ;ISSUE DRIVE CLEAR
 8810  045064  047052                          I.RTRN                  ;ERROR RETURN
 8811  045066  136437  001514  001425          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION RESET
 8812  045074  001407                          BEQ     2$              ;NO, INDICATE DRIVE ERROR
 8813  045076  052737  000020  001462          BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
 8814                                                                  ;  WITH CLEAR
 8815  045104  004777  134350                  JSR     PC,@A.CONT      ;REPORT CONTROLLER ERROR
 8816  045110  000137  047052                  JMP     I.RTRN          ;GO RESTORE REGISTERS
 8817
 8818  045114  032737  040000  001434  2$:     BIT     #S.DSC,T.MR2    ;CHECK IF DRIVE STATUS CHANGE CLEARED
 8819  045122  001403                          BEQ     3$              ;YES, CHECK FAULT
 8820  045124  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DSC DID NOT CLEAR
 8821  045132  032737  001000  001436  3$:     BIT     #S.PAR,T.MR3    ;CHECK IF DRIVE PARITY ERROR
 8822  045140  001407                          BEQ     5$              ;NO, INDICATE ABNORMAL TERMINATION
 8823  045142  052737  002000  001462          BIS     #E.DPAR,E.CONT  ;SET DRIVE PARITY ERROR
 8824  045150  004777  134304                  JSR     PC,@A.CONT      ;INDICATE CONTROLLER ERROR
 8825  045154  000137  047052                  JMP     I.RTRN          ;RETURN
 8826
 8827  045160  032765  000020  000014  5$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF HARD DRIVE ERROR
 8828  045166  001017                          BNE     10$             ;YES, GO REPORT ERROR
 8829  045170  032737  020000  001434          BIT     #S.PIP,T.MR2    ;CHECK IF DRIVE IS CYCLING DOWN
 8830  045176  001413                          BEQ     10$             ;NO, REPORT ERROR
 8831  045200  052765  020000  000014          BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING
 8832  045206  156437  001514  001512          BISB    INTMSK(R4),W.TIME ;SET UP 8 SECONDS FOR
 8833  045214  013763  001474  001544          MOV     W.8SEC,W.DRV(R3) ;  DRIVE TO CYCLE UP
 8834  045222  000137  047052                  JMP     I.RTRN          ;GO RESTORE REGISTERS
 8835
 8836  045226  004777  134224          10$:    JSR     PC,@A.ABNL      ;GO REPORT ERROR
 8837  045232  000137  047052                  JMP     I.RTRN          ;GO RESTORE REGISTERS
 8838
 8839                                  .SBTTL   *READ ALL HEADERS INTERRUPT SEQUENCE
 8840
 8841  045236  016237  000000  001410  I.HDAL: MOV     RKCS1(R2),T.CS1 ;STORE CS1 TO CHECK CONTROLLER
 8842                                                                  ;  ERROR
 8843  045244  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 8844  045252  001427                          BEQ     5$              ;NO, CHECK FOR ATTENTION
```

```
 8845
 8846  045254  005037  001464                    CLR     O.WAIT              ;CLEAR WAITING FOR COMMAND COMPLETE
 8847  045260  146437  001514  001512            BICB    INTMSK(R4),W.TIME ;RESET TIMING ON DRIVE
 8848  045266  005063  001544                    CLR     W.DRV(R3)           ;CLEAR TIME OUT COUNT
 8849  045272  013765  001410  000016            MOV     T.CS1,P.CS1(R5) ;STORE ERROR RKCS1
 8850  045300  004737  047564                    JSR     PC,I.CST1           ;STORE CONTROLLER REGISTERS
 8851  045304  004037  047070                    JSR     R0,I.CCLR           ;CLEAR CONTROLLER
 8852  045310  047052                            I.RTRN                      ;ERROR RETURN
 8853  045312  004777  134140                    JSR     PC,@A.ABNL          ;INDICATE ERROR RETURN
 8854  045316  000137  047052                    JMP     I.RTRN              ;RESTORE REGISTERS
 8855
 8856  045322  016237  000016  001424  5$:       MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 8857  045330  136437  001514  001425            BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS SET
 8858  045336  001411                            BEQ     7$                  ;NO, CHECK IF READ ALL HEADERS
 8859  045340  005037  001464                    CLR     O.WAIT              ;CLEAR WAITING FOR COMMAND COMPLETION
 8860  045344  146437  001514  001512            BICB    INTMSK(R4),W.TIME ;RESET TIMING ON DRIVE
 8861  045352  005063  001544                    CLR     W.DRV(R3)           ;CLEAR TIME OUT COUNT
 8862  045356  000137  044752                    JMP     I.ERRA              ;GO REPORT ERROR
 8863
 8864  045362  013701  001502            7$:     MOV     HDR.AD,R1           ;GET MAIN MEMORY ADDRESS
 8865  045366  016221  000024                    MOV     RKDB(R2),(R1)+      ;GET FIRST WORD OF HEADER
 8866  045372  016221  000024                    MOV     RKDB(R2),(R1)+      ;GET SECOND WORD OF HEADER
 8867  045376  016221  000024                    MOV     RKDB(R2),(R1)+      ;GET THIRD WORD OF HEADER
 8868  045402  010137  001502                    MOV     R1,HDR.AD           ;STORE ADDRESS FOR NEXT HEADER
 8869  045406  016237  000010  001412            MOV     RKCS2(R2),T.CS2 ;STORE CS2 TO CHECK FOR DATA LATE
 8870  045414  032737  100000  001412            BIT     #DLT,T.CS2          ;CHECK FOR DATA LATE
 8871  045422  001056                            BNE     35$                 ;YES, REPORT ERROR
 8872  045424  005337  001504                    DEC     HDR.CT              ;DECREMENT NUMBER OF HEADER YET TO READ
 8873  045430  001027                            BNE     25$                 ;IF NON-ZERO, GO ISSUE NEXT READ HEADER
 8874  045432  005037  001464                    CLR     O.WAIT              ;CLEAR DRIVER WAITING FOR COMMAND COMPLETION
 8875  045436  005063  001544                    CLR     W.DRV(R3)           ;CLEAR TIME OUT COUNT FOR THIS DRIVE
 8876  045442  146437  001514  001512            BICB    INTMSK(R4),W.TIME ;CLEAR WATCH DOG TIME ON THIS DRIVE
 8877  045450  012?02  000003  000026            MOV     #3,RKMR1(R2)        ;LOAD MAINTENANCE REGISTER FOR SECTOR COUNT
 8878  045456  112737  000001  001410            MOVB    #DR.SEL,T.CS1       ;LOAD SELECT COMMAND
 8879  045464  004037  047152                    JSR     R0,I.ISSU           ;GET SECTOR COUNT
 8880  045470  047052                            I.RTRN                      ;ERROR RETURN
 8881  045472  013765  001436  000056            MOV     T.MR3,P.B11(R5) ;LOAD SECTOR COUNT
 8882  045500  004777  133750                    JSR     PC,@A.NORM          ;INDICATE NORMAL TERMINATION
 8883  045504  000137  047052                    JMP     I.RTRN              ;RESTORE REGISTERS
 8884
 8885  045510  016562  000002  000020  25$:     MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
 8886  045516  016562  000004  000006            MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
 8887  045524  116565  000007  000017            MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 8888  045532  042765  165777  000016            BIC     #^C<CDT!CFMT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT AND
 8889                                                                        ;  DRIVE TYPE
 8890  045540  112765  000125  000016            MOVB    #RDHEAD,P.CS1(R5)   ;STORE COMMAND ISSUED
 8891  045546  016562  000016  000000            MOV     P.CS1(R5),RKCS1(R2)  ;ISSUE READ HEADER
 8892  045554  000137  047052                    JMP     I.RTRN              ;RESTORE REGISTERS
 8893
 8894  045560  052737  000400  001462  35$:     BIS     #E.DLT,E.CONT       ;SET DATA LATE WHILE UNLOADING HEADER
 8895  045566  004777  133666                    JSR     PC,@A.CONT          ;REPORT ERROR
 8896  045572  000137  047052                    JMP     I.RTRN              ;RESTORE REGISTERS
 8897
 8898                                    .SBTTL   *DRIVE ATTENTION SCANNER
 8899
 8900  045576  016237  000000  001410  I.ATTN: MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS
```

```
8901                                                    ;  REGISTER 1 FOR COMPARISON
8902  045604  032737  100000  001410        BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCURRED
8903  045612  001444                        BEQ     5$              ;NO, CHECK IF ATTENTION
8904
8905                                        ;CHECK IF ANY DATA TRANSFER TYPE ERROR EXISTS
8906  045614  032737  164000  001412        BIT     #DLT!WCE!UPE!NEM,T.CS2
8907
8908  045622  001007                        BNE     1$              ;INDICATE ERROR
8909  045624  016237  000014  001426        MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
8910
8911                                ;       CHECK FOR DATA TRANSFER ERROR TYPE
8912  045632  032737  125700  001426        BIT     #DCK!OPI!WLE!COE!HVRC!BSE!ECH,T.ER
8913
8914  045640  001407                        BEQ     2$              ;NO DATA TRANSFER ERROR
8915
8916  045642  052737  000010  001462  1$:   BIS     #E.UDAT,E.CONT  ;SET UNEXPECTED DATA TYPE ERROR
8917  045650  004777  133604                JSR     PC,@A.CONT      ;REPORT ERROR
8918  045654  000137  047052                JMP     I.RTRN          ;RESTORE REGISTERS
8919
8920  045660  013704  001412        2$:     MOV     T.CS2,R4        ;SAVE CS2 FOR REGISTER NUMBER
8921  045664  042704  177770                BIC     #^C<DRVMSK>,R4  ;STRIP OFF JUNK
8922  045670  010403                        MOV     R4,R3           ;STORE DRIVE NUMBER IN R3
8923  045672  006303                        ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
8924  045674  146437  001514  001512        BICB    INTMSK(R4),W.TIME ;CLEAR WATCH-DOG TIMER FOR DRIVE
8925  045702  005063  001544                CLR     W.DRV(R3)       ;RESET WATCH DOG TIME
8926  045706  016305  001524                MOV     PBLKT(R3),R5    ;STORE PARAMETER BLOCK ADDRESS IN R5
8927
8928                                ;       CLEAR DRIVE POSITIONING AND DRIVE POSITIONED FOR DATA TRANSFER
8929                                ;         IN PROGRAM DEVICE STATUS REGISTER
8930  045712  042765  000006  000014        BIC     #DRVPOS!DRVPDT,P.PRST(R5)
8931
8932  045720  000137  044760                JMP     I.ERRC          ;GO REPORT ERROR
8933
8934  045724  032737  040000  001410  5$:   BIT     #DI,T.CS1       ;CHECK IF ANY DRIVE ATTENTION
8935  045732  001004                        BNE     6$              ;YES, PROCESS INTERRUPT
8936  045734  004737  052116                JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER
8937  045740  000137  047052                JMP     I.RTRN          ;RESTORE REGISTERS
8938
8939  045744  016237  000016  001424  6$:   MOV     PKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
8940  045752  105737  001425                TSTB    T.ASOF+1        ;CHECK IF ANY ATTENTIONS SET
8941  045756  001007                        BNE     7$              ;YES GO PROCESS INTERRUPT
8942  045760  052737  000002  001462        BIS     #E.NOAT,E.CONT  ;SET NO ATTENTION IN ATTENTION SUMMARY
8943  045766  004777  133466                JSR     PC,@A.CONT      ;GO REPORT ERROR
8944  045772  000137  047052                JMP     I.RTRN          ;GO RESTORE REGISTERS
8945
8946  045776  005004                7$:     CLR     R4              ;CLEAR DRIVE NUMBER
8947  046000  136437  001514  001425  8$:   BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION OF THIS DRIVE
8948  046006  001002                        BNE     9$              ;YES, PROCESS INTERRUPT
8949  046010  005204                        INC     R4              ;INCREMENT DRIVE NUMBER
8950  046012  000772                        BR      8$              ;CHECK ATTENTION ON NEXT DRIVE
8951
8952  046014  010403                9$:     MOV     R4,R3           ;STORE DRIVE NUMBER
8953  046016  006303                        ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
8954  046020  016305  00  >24              MOV     PBLKT(R3),R5    ;STORE PARAMETER BLOCK ADDRESS
8955  046024  032765  0  000  000014        BIT     #DRVSZD,P.PRST(R5) ;CHECK IF DRIVE WAS SEIZED
8956  046032  001402                        BEQ     10$             ;NO, PROCESS NORMAL ATTENTION
```

```
 8957  046034  000137  046664                    JMP     I.DUAL          ;PROCESS THE RELEASE OF DRIVE
 8958
 8959  046040                            10$:
 8960  046040  032765  020000  000014            BIT     #E.UNLD,P.PRST(R5)  ;CHECK IF DRIVE UNLOADING
 8961  046046  001402                            BEQ     11$             ;NO, CONTINUE
 8962  046050  000137  046560                    JMP     I.UNLD          ;SERVICE DRIVE IN POSITION AFTER ERROR
 8963
 8964  046054  042765  000002  000014  11$:     BIC     #DRVPOS,P.PRST(R5)  ;RESET DRIVE POSITIONING
 8965  046062  005062  000026                    CLR     RKMR1(R2)       ;CLEAR MAINTENANCE REGISTER 1
 8966  046066  112737  000001  001410            MOVB    #DR.SEL,T.CS1   ;LOAD COMMAND
 8967  046074  004037  047152                    JSR     R0,I.ISSU       ;SELECT DRIVE WITH ATTENTION HIGH
 8968  046100  047052                            I.RTRN                  ;ERROR RETURN
 8969  046102  013765  001436  000042            MOV     T.MR3,P.B00(R5) ;STORE STATUS BYTE 00 MESS B
 8970  046110  032765  000200  000042            BIT     #S.FLT,P.B00(R5)  ;CHECK IF DRIVE FAULT
 8971  046116  001401                            BEQ     12$             ;NO, CHECK FOR DRIVE STATUS CHANGE
 8972  046120  000473                            BR      I.AERR          ;PROCESS ERROR
 8973
 8974  046122  013765  001434  000040  12$:     MOV     T.MR2,P.A00(R5) ;STORE MAINTENANCE REGISTER 2
 8975  046130  032765  040000  000040            BIT     #S.DSC,P.A00(R5)  ;CHECK FOR DRIVE STATUS CHANGE
 8976  046136  001004                            BNE     13$             ;YES, PROCESS DRIVE STATUS CHANGE
 8977  046140  052765  004000  000014            BIS     #NODSC,P.PRST(R5)  ;SET NO DRIVE STATUS CHANGE
 8978  046146  0004 J                            BR      I.AERR          ;PROCESS ERROR
 8979
 8980  046150  112737  000005  001410  13$:     MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
 8981  046156  004037  047152                    JSR     R0,I.ISSU       ;CLEAR DRIVE STATUS CHANGE
 8982  046162  047052                            I.RTRN                  ;ERROR RETUN
 8983  046164  013765  001424  000032            MOV     T.ASOF,P.ASOF(R5)  ;STORE ATTENTION SUMMARY
 8984  046172  136465  001514  000033            BITB    INTMSK(R4),P.ASOF+1(R5) ;CHECK IF ATTENTION RESET
 8985  046200  001407                            BEQ     15$             ;YES, CONTINUE INTERRUPT PROCESSING
 8986  046202  052737  000020  001462            BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
 8987                                                                    ;  WITH DRIVE CLEAR
 8988  046210  004777  133244                    JSR     PC,@A.CONT      ;FLAG ERROR
 8989  046214  000137  047052                    JMP     I.RTRN          ;RESTORE REGISTERS
 8990
 8991  046220  013765  001434  000040  15$:     MOV     T.MR2,P.A00(R5) ;STORE MAINTENANCE REGISTER 2
 8992  046226  032765  040000  000040            BIT     #S.DSC,P.A00(R5)  ;CHECK IF DRIVE STATUS CHANGE
 8993                                                                    ;  RESET
 8994  046234  001404                            BEQ     16$             ;YES, CONTINUE INTERRUPT PROCESSING
 8995  046236  052765  000040  000014            BIS     #DRVDSC,P.PRST(R5)  ;SET DRIVE STATUS CHANGE DID NOT CLEAR
 8996  046244  000421                            BR      I.AERR          ;GO PROCESS ERROR
 8997
 8998  046246  146437  001514  001512  16$:     BICB    INTMSK(R4),W.TIME  ;RESET TIMING ON THIS DRIVE
 8999  046254  005063  0015?4                    CLR     W.DRV(R3)       ;CLEAR DRIVE TIMING COUNT
 9000  046260  032765  000004  000014            BIT     #DRVPDT,P.PRST(R5)  ;CHECK IF DRIVE IS POSITIONING
 9001                                                                    ;  FOR DATA TRANSFER
 9002  046266  001004                            BNE     17$             ;YES, CALL COMMAND OPTIMIZER
 9003  046270  004777  133160                    JSR     PC,@A.NORM      ;REPORT SUCCESSFUL COMMAND
 9004                                                                    ;  COMPLETION
 9005  046274  000137  047052                    JMP     I.RTRN          ;RESTORE REGISTERS
 9006
 9007  046300  004737  052116          17$:     JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER
 9008  046304  000137  047052                    JMP     I.RTRN          ;RESTORE REGISTERS
 9009
 9010                                    .SBTTL  *ATTENTION ERROR HANDLER
 9011
 9012  046310  042765  000004  000014  I.AERR: BIC     #DRVPDT,P.PRST(R5) ;RESET POSITIONING IN PROGRESS BECAUSE
```

```
                                                        ;  OF DATA TRANSFER
046316 146437 001514 001512         BICB    INTMSK(R4),W.TIME ;CLEAR TIMING FOR THIS DRIVE
046324 005063 001544                CLR     W.DRV(R3)         ;RESET WATCH-DOG TIME
046330 042765 177741 000016         BIC     #177741,P.CS1(R5) ;KEEP COMMAND ISSUED
046336 042737 000036 001410         BIC     #36,T.CS1         ;KEEP CURRENT CONTROLLER STATUS
046344 053765 001410 000016         BIS     T.CS1,P.CS1(R5) ;MAKE GOOD MESSAGE
046352 013765 001412 000020         MOV     T.CS2,P.CS2(R5) ;STORE CONTROLLER REGISTERS
046360 013765 001414 000022         MOV     T.WCR,P.WCR(R5)
046366 013765 001416 000024         MOV     T.BA,P.BAR(R5)
046374 013765 001420 000026         MOV     T.DA,P.DTS(R5)
046402 013765 001422 000030         MOV     T.DC,P.DCYL(R5)
046410 013765 001424 000032         MOV     T.ASOF,P.ASOF(R5)
046416 013765 001426 000034         MOV     T.ER,P.ER(R5)
046424 013765 001430 000036         MOV     T.DS,P.DS(R5)
046432 004037 047646                JSR     R0,I.STAT         ;GATHER DRIVE STATUS
046436 047052                       I.RTRN                    ;ERROR RETURN
046440 112737 000005 001410         MOVB    #DR.CLR,T.CS1     ;LOAD COMMAND
046446 004037 047152                JSR     R0,I.ISSU         ;CLEAR DRIVE ERRORS
046452 047052                       I.RTRN                    ;ERROR RETURN
046454 136437 001514 001425         BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION RESET
046462 001407                       BEQ     2$                ;YES, FLAG DRIVE ERROR
046464 052737 000020 001462         BIS     #E.CLAT,E.CONT    ;SET ATTENTION DID NOT RESET
046472 004777 132762                JSR     PC,@A.CONT        ;REPORT ERROR
046476 000137 047052                JMP     I.RTRN            ;RESTORE REGISTERS

046502 032765 000020 000014 2$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF AHARD DRIVE ERROR
046510 001017                       BNE     10$               ;YES, REPORT ERROR
046512 032737 020000 001434         BIT     #S.PIP,T.MR2      ;CHECK IF DRIVE IS UNLOADING
046520 001413                       BEQ     10$               ;NO, REPORT ERROR
046522 052765 020000 000014         BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING DUE TO ERROR
046530 156437 001514 001512         BISB    INTMSK(R4),W.TIME ;SET TIMING ON THIS DRIVE
046536 013763 001474 001544         MOV     W.8SEC,W.DRV(R3) ;LOAD 8 SECONDS FOR CYCLE UP
046544 000137 047052                JMP     I.RTRN            ;RESTORE REGISTERS

046550 004777 132702          10$:  JSR     PC,@A.ABNL        ;REPORT ERROR
046554 000137 047052                JMP     I.RTRN            ;RESTORE REGISTERS

                              .SBTTL   *ERROR CAUSING DRIVE TO UNLOAD

046560 052765 020000 000014 I.UNLD: BIS     #E.UNLD,P.PRST(R5) ;CLEAR DRIVE UNLOADING BECAUSE OF ERROR
046566 112737 000005 001410         MOVB    #DR.CLR,T.CS1     ;LOAD IN DRIVE CLEAR
046574 004037 047152                JSR     R0,I.ISSU         ;GO ISSUE DRIVE CLEAR
046600 047052                       I.RTRN                    ;ERROR RETURN
046602 136437 001514 001425         BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION CLEARED
046610 001406                       BEQ     15$               ;YES, CONTINUE
046612 012737 000020 001462         MOV     #E.CLAT,E.CONT    ;SET ATTENTION DID NOT RESET
046620 004777 132634                JSR     PC,@A.CONT        ;REPORT ERROR
046624 000512                       BR      I.RTRN            ;RESTORE REGISTERS

046626 032737 040000 001434 15$:    BIT     #S.DSC,T.MR2      ;CHECK IF DRIVE STAUS CHANGE RESET
046634 001403                       BEQ     20$               ;YES, CONTINUE
046636 052765 000040 000014         BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STAUS CHANGE DID NOT CLEAR
046644 146437 001514 001512 20$:    BICB    INTMSK(R4),W.TIME ;RESET TIMING ON THIS DRIVE
046652 005063 001544                CLR     W.DRV(R3)         ;CLEAR TIME COUNT
046656 004777 132574                JSR     PC,@A.ABNL        ;REPORT ERROR
046662 000473                       BR      I.RTRN            ;RESTORE REGISTER
```

```
                                        .SBTTL   *DUAL ACCESS INTERRUPT HANDLER

046664  112737  000001  001410  I.DUAL: MOVB    #DR.SEL,T.CS1   ;LOAD COMMAND
046672  004037  047152                  JSR     R0,I.ISSU       ;SELECT DRIVE
046676  047052                          I.RTRN                  ;ERROR RETURN
046700  032737  000200  001436          BIT     #S.FLT,T.MR3    ;CHECK IF DRIVE FAULT
046706  001402                          BEQ     10$             ;NO, PROCESS INTERRUPT
046710  000137  046310                  JMP     I.AERR          ;PROCESS ATTENTION ERROR

046714  032737  040000  001434  10$:    BIT     #S.DSC,T.MR2    ;CHECK IF DRIVE STATUS CHANGE SET
046722  001005                          BNE     11$             ;YES, PROCESS INTERRUPT
046724  052765  004000  000014          BIS     #NODSC,P.PRST(R5) ;SET NO DRIVE STATUS CHANGE
046732  000137  046310                  JMP     I.AERR          ;PROCESS ATTENTION ERROR

046736  042765  010000  000014  11$:    BIC     #DRVSZD,P.PRST(R5) ;RESET DRIVE SEIZED
046744  112737  000005  001410          MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
046752  004037  047152                  JSR     R0,I.ISSU       ;CLEAR DRIVE DSC
046756  047052                          I.RTRN                  ;ERROR RETURN
046760  136437  001514  001425          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION RESET
046766  001406                          BEQ     15$             ;YES, CONTINUE
046770  052737  000020  001462          BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
046776  004777  132456                  JSR     PC,@A.CONT      ;INDICATE CONTROLLER ERROR
047002  000423                          BR      I.RTRN          ;RESTORE REGISTERS

047004  032737  040000  001434  15$:    BIT     #S.DSC,T.MR2    ;CHECK IF DRIVE STATUS CHANGE IS STILL SET
047012  001405                          BEQ     20$             ;NO, GO ENQUEUE COMMAND
047014  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STATUS CHANGE DID NOT CLEAR
047022  000137  046310                  JMP     I.AERR          ;REPORT ATTENTION ERROR

047026  146437  001514  001512  20$:    BICB    INTMSK(R4),W.TIME ;STOP TIMING ON THIS DRIVE
047034  005063  001544                  CLR     W.DRV(R3)       ;CLEAR WATCH DOG TIME OF THIS DRIVE
04704C  004037  051564                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ADDRESS ON
047044  001324                          CINITQ                  ;  COMMAND INITATION QUEUE
047046  004737  052116                  JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER

047052  012600          I.RTRN: MOV     (SP)+,R0        ;RESTORE R0
047054  012601                  MOV     (SP)+,R1        ;RESTORE R1
047056  012602                  MOV     (SP)+,R2        ;RESTORE R2
047060  012603                  MOV     (SP)+,R3        ;RESTORE R3
047062  012604                  MOV     (SP)+,R4        ;RESTORE R4
047064  012605                  MOV     (SP)+,R5        ;RESTORE R5
047066  000002                  RTI                     ;RETURN
```

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

047070  012762  100000  000000  I.CCLR: MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
047076  016237  000000  001410          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
047104  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER CLEAR DID
                                                                ;  CLEAR ERROR
047112  001407                          BEQ     5$              ;YES, RETURN TO DRIVER PROCESSING
047114  052737  000001  001462          BIS     #E.CCLR,E.CONT  ;SET CLEAR CONTROLLER DID NOT CLEAR ERROR
047122  004777  132332                  JSR     PC,@A.CONT      ;REPORT CONTROLLER ERROR
047126  011000                          MOV     (R0),R0         ;SET UP ERROR RETURN
047130  000200                          RTS     R0              ;RETURN

047132  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
047140  112737  177777  001506          MOVB    #-1,I.ISRL      ;SET INTERRUPT ENABLE ISSUED
047146  005720                          TST     (R0)+           ;ADJUST FOR NORMAL RETURN
047150  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
                                ;*
                                ;;*****************************************************************

047152  013746  001410          I.ISSU: MOV     T.CS1,-(SP)     ;STORE COMMAND ISSUED
047156  005037  001412                  CLR     T.CS2           ;CLEAR TEMPORARY CS2
047162  116537  000000  001412          MOVB    P.DRVN(R5),T.CS2 ;LOAD IN DRIVE NUMBER
047170  013762  001412  000010          MOV     T.CS2,RKCS2(R2) ;LOAD DRIVE NUMBER FOR COMMAND
047176  116537  000007  001411          MOVB    P.CS1H(R5),T.CS1+1 ;STORE BITS 8-15 OF CS1
047204  142737  177753  001411          BICB    #^C<B.CDT!B.CFMT>,T.CS1+1 ;CLEAR ALL BITS EXCEPT
                                                                ;  FORMAT AND DRIVE TYPE
047212  013762  001410  000000          MOV     T.CS1,RKCS1(R2) ;ISSUE COMMAND
047220  105762  000000          1$:     TSTB    RKCS1(R2)       ;WAIT FOR READY
047224  100375                          BPL     1$
047226  004737  047414                  JSR     PC,I.STOR       ;GO STORE REGISTERS
047232  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCUERED
047240  001447                          BEQ     5$              ;NO, RETURN
047242  032737  001000  001412          BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
047250  001406                          BEQ     2$              ;NO, CHECK FOR OTHER CONTROLLER ERRORS
047252  052737  100000  001462          BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT FLAG
047260  004777  132174                  JSR     PC,@A.CONT      ;REPORT CONTROLLER ERROR
047264  000450                          BR      10$             ;RETURN

                                        ;CHECK IF ANY CONTROLLER ERROR IS SET
047266  022737  000001  001500  2$:     CMP     #1,OPTFLG       ;SEE IF CURRENTLY IN CMND OPTIMIZER
047274  001004                          BNE     4$              ;BR IF NOT
047276  032737  010000  001412          BIT     #NED,T.CS2      ;SEE IF NED ERROR SET
047304  001022                          BNE     3$              ;BR IF YES, TO ISSUE CLEAR AND EXIT
047306  032737  024000  001410  4$:     BIT     #CTO!SPAR,T.CS1
047314  001027                          BNE     7$
047316  032737  176400  001412          BIT     #UFE!PGE!NEM!NED!UPE!WCE!DLT,T.CS2
047324  001023                          BNE     7$
```

```
 9204  047326  032737  131761  001426          BIT     #ILC!DTYE!FMTE!ECH!BSE!HVRC!COE!DTE!OPI!DCK,T.ER
 9205  047334  001017                          BNE     7$
 9206
 9207  047336  122716  000005                  CMPB    #DR.CLR,(SP)    ;CHECK IF CLEAR DRIVE
 9208  047342  001003                          BNE     3$              ;NO, DO NOT SET DRIVE HARD ERROR
 9209  047344  052765  000020  000014          BIS     #DRVHRD,P.PRST(R5) ;SET HARD DRIVE ERROR
 9210  047352  004037  047070          3$:     JSR     R0,I.CCLR       ;GO ISSUE A CONTROLLER CLEAR
 9211  047356  047406                          10$                     ;ERROR RETURN
 9212  047360  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
 9213  047366  005726                          TST     (SP)+           ;ADJUST STACK
 9214  047370  005720                          TST     (R0)+           ;ADJUST R0 FOR NORMAL RETURN
 9215  047372  000200                          RTS     R0              ;RETURN
 9216
 9217  047374  052737  001000  001462  7$:     BIS     #E.CERR,E.CONT  ;SET CONTROLLER ERROR DURING
 9218                                                                  ;  DRIVER SERVICING
 9219  047402  004777  132052                  JSR     PC,@A.CONT      ;REPORT ERROR
 9220  047406  005726                  10$:    TST     (SP)+           ;ADJUST STACK
 9221  047410  011000                          MOV     (R0),R0         ;ADJUST R0 FOR ERROR RETURN
 9222  047412  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
                        ;*
                        ;;*****************************************************************

047414  016237  000000  001410  I.STOR: MOV     RKCS1(R2),T.CS1 ;STORE ALL CONTROLLER REGISTERS
047422  016237  000010  001412          MOV     RKCS2(R2),T.CS2 ;  EXCEPT DATA BUFFER
047430  016237  000002  001414          MOV     RKWC(R2),T.WCR
047436  016237  000004  001416          MOV     RKBA(R2),T.BA
047444  016237  000006  001420          MOV     RKDA(R2),T.DA
047452  016237  000012  001430          MOV     RKDS(R2),T.DS
047460  016237  000014  001426          MOV     RKER(R2),T.ER
047466  016237  000016  001424          MOV     RKASOF(R2),T.ASOF
047474  016237  000020  001422          MOV     RKDCYL(R2),T.DC
047502  016237  000026  001432          MOV     RKMR1(R2),T.MR1
047510  016237  000034  001434          MOV     RKMR2(R2),T.MR2
047516  016237  000036  001436          MOV     RKMR3(R2),T.MR3
047524  016237  000030  001440          MOV     RKECPS(R2),T.POS
047532  016237  000032  001442          MOV     RKECPT(R2),T.PAT
047540  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
                                ;*
                                ;;*****************************************************************

 047542  042765  177741  000016  I.CSTS: BIC     #177741,P.CS1(R5)       ;CLEAR ALL BITS EXCEPT FUNCTION
                                                                         ;OF LAST COMMAND ISSUED
 047550  042737  000036  001410          BIC     #36,T.CS1               ;CLEAR FUNCTION OF CS1 STATUS
 047556  053765  001410  000016          BIS     T.CS1,P.CS1(R5)         ;GENERATE CS1 STATUS INFORMATION
 047564  016265  000010  000020  I.CST1: MOV     RKCS2(R2),P.CS2(R5)     ;STORE COMMAND AND STATUS REGISTER 2
 047572  016265  000002  000022          MOV     RKWC(R2),P.WCR(R5)      ;STORE WORD COUNT REGISTER
 047600  016265  000004  000024          MOV     RKBA(R2),P.BAR(R5)      ;STORE BUS ADDRESS REGISTER
 047606  016265  000006  000026          MOV     RKDA(R2),P.DTS(R5)      ;STORE DESIRED TRACK AND SECTOR
 047614  016265  000012  000036          MOV     RKDS(R2),P.DS(R5)       ;STORE DRIVE STATUS REGISTER
 047622  016265  000014  000034          MOV     RKER(R2),P.ER(R5)       ;STORE ERROR REGISTER
 047630  016265  000016  000032          MOV     RKASOF(R2),P.ASOF(R5)   ;STORE ATTENTION SUMMARY AND
                                                                         ;  OFFSET
 047636  016265  000020  000030          MOV     RKDCYL(R2),P.DCYL(R5)   ;STORE CYLINDER ADDRESS
 047644  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
                        ;*
                        ;;*****************************************************************
047646  012762  000001  000026  I.STAT: MOV     #1,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 01
047654  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047662  004037  047152                  JSR     R0,I.ISSU               ;GET STATUS BYTES 01
047666  050056                          3$                              ;ERROR RETURN
047670  013765  001434  000044          MOV     T.MR2,P.A01(R5)         ;STORE STATUS BYTE 01 MESS A
047676  013765  001436  000046          MOV     T.MR3,P.B01(R5)         ;STORE STATUS BYTE 01 MESS B
047704  012762  000002  000026          MOV     #2,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 10
047712  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047720  004037  047152                  JSR     R0,I.ISSU               ;GET STATUS BYTES 10
047724  050056                          3$                              ;ERROR RETURN
047726  013765  001434  000050          MOV     T.MR2,P.A10(R5)         ;STORE STATUS BYTE 10 MESS A
047734  013765  001436  000052          MOV     T.MR3,P.B10(R5)         ;STORE STATUS BYTE 10 MESS B
047742  012762  000003  000026          MOV     #3,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER
                                                                        ;  FOR STATUS BYTE 11
047750  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047756  004037  047152                  JSR     R0,I.ISSU               ;GET STATUS BYTES 11
047762  050056                          3$                              ;ERROR RETURN
047764  013765  001434  000054          MOV     T.MR2,P.A11(R5)         ;STORE STATUS BYTE 11 MESS A
047772  013765  001436  000056          MOV     T.MR3,P.B11(R5)         ;STORE STATUS BYTE 11 MESS B
050000  005062  000026                  CLR     RKMR1(R2)               ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 00
050004  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
050012  004037  047152                  JSR     R0,I.ISSU               ;GET STATUS BYTES 00
050016  050056                          3$                              ;ERROR RETURN
050020  013765  001434  000040          MOV     T.MR2,P.A00(R5)         ;STORE STATUS BYTE 00 MESS A
050026  013765  001436  000042          MOV     T.MR3,P.B00(R5)         ;STORE STATUS BYTE 00 MESS B
050034  032737  001000  001436          BIT     #S.PAR,T.MR3            ;CHECK IF BAD PARITY DETECTED BY DRIVE
050042  001407                          BEQ     5$                      ;NO, RETURN NORMALLY
```

```
 9356  050044  052737  002000  001462          BIS     #E.DPAR,E.CONT          ;INDICATE BAD PARITY DETECTED BY DRIVE
 9357  050052  004777  131402                  JSR     PC,@A.CONT              ;REPORT ERROR
 9358  050056  011000                  3$:     MOV     (R0),R0                 ;LOAD R0 FOR ERROR RETURN
 9359  050060  000200                          RTS     R0                      ;RETURN
 9360
 9361  050062  052765  001000  000( 14  5$:     BIS     #PBSVAL,P.PRST(R5)      ;SET PARAMETER BLOCK STATUS VALID
 9362  050070  005720                          TST     (R0)+                   ;ADJUST R0 FOR NORMAL RETURN
 9363  050072  000200                          RTS     R0                      ;RETURN
 9364
```

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

050074  010546          C.INIT: MOV     R5,-(SP)        ;STORE R5 ON STACK
050076  010446                  MOV     R4,-(SP)        ;STORE R4 ON STACK
050100  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
050102  010246                  MOV     R2,-(SP)        ;STORE R2 ON STACK
050104  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
050106  010046                  MOV     R0,-(SP)        ;STORE R0 ON STACK
050110  013746  177776          MOV     PS,-(SP)        ;STORE PSW ON STACK
050114  013737  001452  177776  MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
050122  017605  000016          MOV     @16(SP),R5      ;STORE PARAMETER BLOCK ADDRESS
050126  062766  000002  000016  ADD     #2,16(SP)       ;ADJUST RETURN
050134  016504  000000          MOV     P.DRVN(R5),R4   ;STORE DRIVE NUMBER
050140  042704  177770          BIC     #^C<DRVMSK>,R4  ;MASK OUT JUNK
050144  156437  001514  001512  BISB    INTMSK(R4),W.TIME ;SET WATCH DOG TIME
050152  010403                  MOV     R4,R3           ;STORE DRIVE NUMBER
050154  006303                  ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
050156  013763  001472  001544  MOV     W.SEC,W.DRV(R3) ;LOAD WATCH-DOG TIME

050164  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
```

```
 9421  050170  042765  075176  000014          BIC     #^C<DRVUSE!W.WCK!NOCHK!DRPDRV!DTBAII>,P.PRST(R5)
 9422
 9423  050176  010500                          MOV     R5,R0           ;STORE PARAMETER BLOCK ADDRESS
 9424  050200  062700  000016                  ADD     #P.CS1,R0       ;CALCULATE FIRST LOCATION TO BE CLEARED
 9425  050204  010501                          MOV     R5,R1           ;STORE PARAMETER BLOCK ADDRESS
 9426  050206  062701  000062                  ADD     #P.EPAT,R1      ;CALCULATE LAST LOCATION TO BE CLEARED
 9427
 9428  050212  005020                  1$:     CLR     (R0)+           ;CLEAR RETURN PARAMETER
 9429  050214  020001                          CMP     R0,R1           ;CHECK IF FINISHED
 9430  050216  101775                          BLOS    1$              ;NO, CLEAR NEXT RETURN PARAMETER
 9431  050220  105037  001506                  CLRB    I.ISRL          ;CLEAR RELEASE OR INTERRUPT ISSUED
 9432  050224  010465  000020                  MOV     R4,P.CS2(R5)    ;STORE DRIVE NUMBER
 9433  050230  005062  000026                  CLR     RKMR1(R2)       ;CLEAR RK06 MAINTENANCE REGISTER 1
 9434  050234  132765  000040  000001          BITB    #BIT5,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
 9435  050242  001402                          BEQ     3$              ;NO, PROCESS
 9436  050244  000137  050624                  JMP     C.SPEC          ;JUMP TO SPECIAL COMMAND PROCESSOR
 9437
 9438  050250  122765  000107  000001  3$:     CMPB    #UNLOAD,P.CMND(R5) ;CHECK IF POSITIONING COMMAND
 9439                                                                  ;  START SPINDLE
 9440                                                                  ;  RECALIBRATE
 9441                                                                  ;  OFFSET
 9442                                                                  ;  SEEK
 9443                                                                  ;  UNLOAD
 9444
 9445  050256  101125                          BHI     25$             ;NO, DRIVE COMMAND
 9446                                                                  ;  SELECT DRIVE
 9447                                                                  ;  PACK ACKNOWLEDGE
 9448                                                                  ;  CLEAR
 9449
 9450  050260  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF DATA TRANSFER
 9451  050266  103471                          BLO     20$             ;YES, DATA TRANSFER COMMAND
 9452                                                                  ;  READ DATA
 9453                                                                  ;  WRITE DATA
 9454                                                                  ;  READ HEADER
 9455                                                                  ;  WRITE HEADER
 9456                                                                  ;  WRITE CHECK
 9457  050270  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
 9458  050276  052765  000002  000014          BIS     #DRVPOS,P.PRST(R5) ;SET DRIVE POSITIONING
 9459  050304  005037  001464                  CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND
 9460  050310  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK
 9461  050316  001007                          BNE     5$              ;NO, CHECK FOR OFFSET
 9462  050320  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 9463  050326  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
 9464  050334  000431                          BR      8$              ;GO ISSUE COMMAND
 9465
 9466  050336  122765  000115  000001  5$:     CMPB    #OFFSET,P.CMND(R5) ;CHECK IF OFFSET
 9467  050344  001007                          BNE     6$              ;NO, CHECK FOR UNLOAD
 9468  050346  116565  000006  000032          MOVB    P.OFST(R5),P.ASOF(R5) ;STORE OFFSET
 9469  050354  016562  000032  000016          MOV     P.ASOF(R5),RKASOF(R2) ;LOAD OFFSET REGISTER
 9470  050362  000416                          BR      8$              ;GO ISSUE COMMAND
 9471
 9472  050364  122765  000111  000001  6$:     CMPB    #SRTSPL,P.CMND(R5) ;CHECK IF START SPINDLE
 9473  050372  001003                          BNE     7$              ;NO, CHECK IF RECAL
 9474  050374  013763  001476  001544          MOV     W.MIN,W.DRV(R3) ;LOAD WATCH DOG TIME FOR 1 MINUTE
 9475  050402  122765  000113  000001  7$:     CMPB    #RECAL,P.CMND(R5) ;CHECK IF RECAL
 9476  050410  001003                          BNE     8$              ;NO, CONTINUE
```

```
 9477  050412  013763  001474  001544          MOV     W.8SEC,W.DRV(R3) ;LOAD RECAL TIME TO 8 SECONDS
 9478  050420  116565  000007  000017  8$:     MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9479  050426  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 9480                                                                  ;  AND DRIVE TYPE
 9481  050434  116565  000001  000016          MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 9482  050442  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 9483  050450  000460                          BR      30$             ;GO, ISSUE COMMAND
 9484
 9485  050452  016562  000010  000004  20$:     MOV     P.BALO(R5),RKBA(R2) ;LOAD BUS ADDRESS REGISTER
 9486  050460  016562  000012  000002          MOV     P.WC(R5),RKWC(R2) ;LOAD WORD COUNT REGISTER
 9487  050466  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
 9488  050474  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK NUMBER
 9489  050502  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)  ;CHECK IF WRITE CHECK COMMAND
 9490  050510  001010                          BNE     25$             ;NO, GO ISSUE THE COMMAND
 9491  050512  032765  000200  000014          BIT     #W.WCK,P.PRST(R5)  ;CHECK IF WRITE COMMAND SHOULD BF ISSUED
 9492  050520  001404                          BEQ     25$             ;NO, GO ISSUE THE COMMAND
 9493  050522  012765  000123  000016          MOV     #WRDATA,P.CS1(R5)  ;ISSUE WRITE COMMAND
 9494  050530  000406                          BR      26$             ;GO ISSUE COMMAND
 9495
 9496  050532  116565  000001  000016  25$:     MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 9497  050540  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 9498  050546  116565  000007  000017  26$:     MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9499  050554  142765  177750  000017          BICB    #^C<B.CFMT!B.CDT!B.BA16!B.BA17>,P.CS1+1(R5) ;CLEAR ALL BITS EXCEPT
 9500                                                                  ;  FORMAT, DRIVE TYPE, AND BUS ADDRESS
 9501                                                                  ;  BITS 16-17
 9502  050562  010537  001464                  MOV     R5,0.WAIT       ;LOAD WAITING FOR COMMAND
 9503  050566  032765  100000  000014          BIT     #DTBAII,P.PRST(R5) ;CHECK IF INHIBIT BUS ADDRESS INCREMENT
 9504  050574  001403                          BEQ     27$             ;NO, LOAD CS2
 9505  050576  052765  000020  000020          BIS     #BAI,P.CS2(R5)  ;SET INHIBIT BUS ADDRESS INCREMENT
 9506  050604  016562  000020  000010  27$:     MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2
 9507  050612  016562  000016  000000  30$:     MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 9508  050620  000137  051542                  JMP     C.RTRN          ;RESTORE REGISTERS
 9509
 9510                                  .SBTTL   *SPECIAL COMMAND PROCESSING
 9511
 9512  050624  122765  000141  000001  C.SPEC: CMPB    #RDSTAT,P.CMND(R5) ;CHECK IF READ DRIVE STATUS
 9513  050632  001132                          BNE     10$             ;NO, PROCESS OTHER COMMANDS
 9514  050634  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR COMMAND
 9515  050642  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9516  050650  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 9517                                                                  ;  AND DRIVE TYPE
 9518  050656  112765  000001  000016          MOVB    #DR.SEL,P.CS1(R5) ;STORE COMMAND
 9519  050664  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 9520  050672  004737  044270          2$:     JSR     PC,W.WTCH       ;CALL WATCH-DOG TIMER
 9521  050676  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REG. 1
 9522  050704  032765  000200  000016          BIT     #RDY,P.CS1(R5)  ;WAIT FOR READY
 9523  050712  001767                          BEQ     2$
 9524  050714  004737  047564                  JSR     PC,I.CST1       ;STORE CONTROLLER REGISTERS
 9525  050720  016265  000034  000040          MOV     RKMR2(R2),P.A00(R5) ;STORE STATUS BYTE 00 MESSAGE A
 9526  050726  016265  000036  000042          MOV     RKMR3(R2),P.B00(R5) ;STORE STATUS BYTE 00 MESSAGE B
 9527  050734  032765  100000  000016          BIT     #CERR,P.CS1(R5) ;CHECK IF CONTROLLER ERROR
 9528  050742  001443                          BEQ     6$              ;NO, GATHER DRIVE STATUS
 9529  050744  146437  001514  001512          BICB    INTMSK(R4),W.TIME ;RESET WATCH DOG TIMING ON THIS DRIVE
 9530  050752  005063  001544                  CLR     W.DRV(R3)       ;CLEAR WATCH DOG COUNT
 9531  050756  032765  001000  000020          BIT     #MDS,P.CS2(R5)  ;CHECK IF MULTIPLE DRIVE SELECT
 9532  050764  001046                          BNE     9$              ;YES, INDICATE CONTROLLER ERROR
```

```
 9533  050766  004037  047070                    JSR     R0,I.CCLR         ;CLEAR ERROR
 9534  050772  051542                            C.RTRN                    ;ERROR RETURN
 9535  050774  032765  010400  000020            BIT     #NED!UFE,P.CS2(R5) ;CHECK IF NON-EXISTENT DRIVE OR UNIT FIELD ERROR
 9536  051002  001017                            BNE     5$                ;REPORT ERROR
 9537  051004  032765  000001  000036            BIT     #DRA,P.DS(R5)     ;CHECK IF DRIVE AVAILIABLE
 9538  051012  001013                            BNE     5$                ;YES, REPORT ERROR
 9539  051014  156437  001514  001512            BISB    INTMSK(R4),W.TIME ;SET TIMING FOR THIS DRIVE
 9540  051022  013763  001476  001544            MOV     W.MIN,W.DRV(R3) ;WAIT 1 MINUTE FOR RELEASE
 9541  051030  052765  010000  000014            BIS     #DRVSZD,P.PRST(R5) ;SET DRIVE SEIZED
 9542  051036  000137  051542                    JMP     C.RTRN            ;RESTORE REGISTERS
 9543
 9544  051042  004777  130410          5$:     JSR     PC,@A.ABNL        ;REPORT ERROR
 9545  051046  000137  051542                    JMP     C.RTRN            ;RESTORE REGISTERS
 9546
 9547  051052  004037  047646          6$:     JSR     R0,I.STAT         ;GATHER DRIVE STATUS
 9548  051056  051542                            C.RTRN                    ;ERROR RETURN
 9549  051060  146437  001514  001512            BICB    INTMSK(R4),W.TIME ;STOP WATCH-DOG TIMING ON DRIVE
 9550  051066  005063  001544                    CLR     W.DRV(R3)         ;RESET WATCH-DOG TIME
 9551  051072  004777  130356                    JSR     PC,@A.NORM        ;REPORT COMMAND COMPLETE
 9552  051076  000137  051542                    JMP     C.RTRN            ;RESTORE REGISTERS
 9553
 9554  051102  052737  100000  001462  9$:     BIS     #E.MDS,E.CONT     ;SET MULTIPLE DRIVE SELECT
 9555  051110  004777  130344                    JSR     PC,@A.CONT        ;INDICATE CONTROLLER ERROR
 9556  051114  000137  051542                    JMP     C.RTRN
 9557
 9558  051120  122765  000140  000001  10$:     CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE COMMAND
 9559  051126  001031                            BNE     13$               ;NO, CHECK IF READ ALL HEADERS
 9560  051130  010537  001464                    MOV     R5,O.WAIT         ;STORE PARAMETER BLOCK ADDRESS IN
 9561                                                                      ;  WAIT FOR COMMAND
 9562  051134  052765  000010  000020            BIS     #RLS,P.CS2(R5)    ;SET RELEASE BIT
 9563  051142  016562  000020  000010            MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR DESELECT
 9564  051150  112737  000001  001506            MOVB    #1,I.ISRL         ;SET FLAG FOR RELEASE COMMAND
 9565  051156  116565  000007  000017            MOVB    P.CS1H(R5),P.CS1+1(R5);STORE BITS 8-15 OF CS1
 9566  051164  042765  165777  000016            BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 9567                                                                      ;  AND DRIVE TYPE
 9568  051172  112765  000101  000016            MOVB    #SELDRV,P.CS1(R5) ;STORE COMMAND
 9569  051200  016562  000016  000000  11$:     MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 9570  051206  000137  051542                    JMP     C.RTRN            ;RESTORE REGISTERS
 9571
 9572  051212  122765  000164  000001  13$:     CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
 9573  051220  001047                            BNE     30$               ;NO, CHECK IF CONTROLLER CLEAR
 9574  051222  010537  001464                    MOV     R5,O.WAIT         ;SET WAITING FOR COMMAND COMMPLETION
 9575  051226  016537  000010  001502            MOV     P.BALO(R5),HDR.AD ;LOAD HEADER ADDRESS
 9576  051234  132765  000020  000007            BITB    #8.CFMT,P.CS1H(R5) ;CHECK IF 22 SECTOR FORMANT
 9577  051242  001404                            BEQ     14$               ;YES, LOAD 22 IN HEADER COUNT
 9578  051244  012737  000024  001504            MOV     #20.,HDR.CT       ;LOAD 20 IN SECTOR COUNT
 9579  051252  000403                            BR      22$               ;GO ISSUE READ HEADER COMMAND
 9580
 9581  051254  012737  000026  001504  14$:     MOV     #22.,HDR.CT       ;LOAD 22 IN SECTOR COUNT
 9582  051262  016562  000002  000020  22$:     MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 9583  051270  016562  000004  000006            MOV     P.SECT(R5),RKDA(R2) ;LOAD TRACK NUMBER
 9584  051276  016562  000020  000010            MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
 9585  051304  116565  000007  000017            MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9586  051312  042765  165777  000016            BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT DRIVE TYPE
 9587                                                                      ;  AND FORMAT
 9588  051320  112765  000125  000016            MOVB    #RDHEAD,P.CS1(R5) ;STORE READ HEADER COMMAND
```

```
 9589  051326  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE READ HEADER
 9590  051334  000137  051542                  JMP     C.RTRN              ;RESTORE REGISTERS
 9591
 9592  051340  122765  000176  000001  30$:    CMPB    #CONCLR,P.CMND(R5) ;CHECK IF CONTROLLER CLEAR
 9593  051346  001004                          BNE     32$                 ;NO, CHECK IF SUBSYSTEM CLEAR
 9594  051350  004037  047070                  JSR     R0,I.CCLR           ;CLEAR CONTROLLER
 9595  051354  051542                          C.RTRN                      ;ERROR RETURN
 9596  051356  000467                          BR      40$                 ;INDICATE NORMAL RETURN
 9597
 9598  051360  122765  000177  000001  32$:    CMPB    #SUBCLR,P.CMND(R5) ;CHECK IF SUBSYSTEM CLEAR
 9599  051366  001406                          BEQ     36$                 ;YES, CLEAR SUBSYSTEM
 9600  051370  052737  000100  001462  34$:    BIS     #E.ILLD,E.CONT      ;SET ILLEGAL DRIVER COMMAND
 9601  051376  004777  130056                  JSR     PC,@A.CONT          ;REPORT ERROR
 9602  051402  000457                          BR      C.RTRN              ;RESTORE REGISTERS
 9603
 9604  051404  012762  000040  000010  36$:    MOV     #SCLR,RKCS2(R2) ;ISSUE SUBSYSTEM CLEAR
 9605  051412  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REGISTER 1
 9606  051420  032765  100000  000016          BIT     #CERR,P.CS1(R5) ;CLEAR IF CONTROLLER ERROR RESET
 9607  051426  001406                          BEQ     37$                 ;NO, FINISH COMMAND
 9608  051430  052737  000001  001462          BIS     #BIT0,E.CONT        ;SET CLEAR SUBSYSTEM DID NOT CLEAR
 9609                                                                      ;  CONTROLLER ERROR
 9610  051436  004777  130016                  JSR     PC,@A.CONT          ;REPORT ERROR
 9611  051442  000437                          BR      C.RTRN              ;RESTORE REGISTERS
 9612
 9613  051444  013746  001470          37$:    MOV     W.MILI,-(SP)        ;LOAD 16 MILI-SECOND COUNT FOR ATTENTION
 9614                                                                      ;  TO DISAPPEAR
 9615  051450  016265  000000  000016  38$:    MOV     RKCS1(R2),P.CS1(R5) ;STORE CS1
 9616  051456  032765  040000  000016          BIT     #DI,P.CS1(R5)       ;CHECK IF ATTENTIONS CLEARED
 9617  051464  001411                          BEQ     39$                 ;YES, FINISH COMMAND
 9618  051466  005316                          DEC     (SP)                ;DECREMENT 16 MILISECOND COUNT
 9619  051470  001367                          BNE     38$                 ;CHECK DRIVE INTERRUPT AGAIN
 9620  051472  005726                          TST     (SP)+               ;ADJUST STACK
 9621  051474  052737  000040  001462          BIS     #E.SCLR,E.CONT      ;SET SUBSYSTEM CLEAR DID NOT CLEAR
 9622                                                                      ;  DRIVE ATTENTIONS
 9623  051502  004777  127752                  JSR     PC,@A.CONT          ;REPORT ERROR
 9624  051506  000415                          BR      C.RTRN              ;RESTORE REGISTER
 9625
 9626  051510  005726                  39$:    TST     (SP)+               ;ADJUST STACK
 9627  051512  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 9628  051520  001010                          BNE     C.RTRN              ;YES, RESTORE REGISTERS
 9629  051522  112737  177777  001506          MOVB    #-1,I.ISRL          ;SET INTERRUPT ENABLE SET
 9630  051530  012762  000100  000000          MOV     #IE,RKCS1(R2)       ;SET INTERRUPT ENABLE
 9631  051536  004777  127712          40$:    JSR     PC,@A.NORM          ;INDICATE NORMAL TERMINATION
 9632
 9633  051542  012637  177776          C.RTRN: MOV     (SP)+,PS            ;RESTORE PSW
 9634  051546  012600                          MOV     (SP)+,R0            ;RESTORE R0
 9635  051550  012601                          MOV     (SP)+,R1            ;RESTORE R1
 9636  051552  012602                          MOV     (SP)+,R2            ;RESTORE R2
 9637  051554  012603                          MOV     (SP)+,R3            ;RESTORE R3
 9638  051556  012604                          MOV     (SP)+,R4            ;RESTORE R4
 9639  051560  012605                          MOV     (SP)+,R5            ;RESTORE R5
 9640  051562  000207                          RTS     PC                  ;RETURN
```

```
                                .SBTTL   *ENQUEUE ROUTINE

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL BUT THE ADDRESS OF THE PARAMETER IN THE
                                ;*      QUEUE SPECIFIED BY THE LOCATION FOLLOWING THE CALL.
                                ;*
                                ;*CALL  JSR     R0,Q.PUSH
                                ;*      <ADDRESS OF QUEUE>
                                ;*      RETURN
                                ;*
                                ;;*****************************************************************

 051564  010446                 Q.PUSH: MOV     R4,-(SP)        ;STORE R4 ON STACK
 051566  010346                         MOV     R3,-(SP)        ;STORE R3 ON STACK
 051570  013746  177776                 MOV     PS,-(SP)        ;STORE PSW ON STACK
 051574  013737  001452  177776         MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 051602  012004                         MOV     (R0)+,R4        ;STORE QUEUE ADDRESS
 051604  005065  000064                 CLR     P.QLNK(R5)      ;CLEAR QUEUE LINK
 051610  005724                         TST     (R4)+           ;CHECK IF QUEUE IS EMPTY
 051612  001405                         BEQ     1$              ;YES, PUT PARAMETER BLOCK ADDRESS
                                                                ;  AT HEAD AND TAIL OF THE QUEUE
 051614  011403                         MOV     (R4),R3         ;STORE TAIL OF QUEUE
 051616  010563  000064                 MOV     R5,P.QLNK(R3)   ;UPDATE QUEUE LINK
 051622  010514                         MOV     R5,(R4)         ;UPDATE QUEUE TAIL
 051624  000402                         BR      2$

 051626  010514                 1$:     MOV     R5,(R4)         ;LOAD TAIL OF QUEUE
 051630  010544                         MOV     R5,-(R4)        ;LOAD HEAD OF QUEUE

 051632  012637  177776         2$:     MOV     (SP)+,PS        ;RESTORE PSW
 051636  012603                         MOV     (SP)+,R3        ;RESTORE R3
 051640  012604                         MOV     (SP)+,R4        ;RESTORE R4
 051642  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
                        ;*
                        ;;*****************************************************************

051644  011646          Q.POP:  MOV     (SP),-(SP)      ;CREATE SPACE ON THE STACK
051646  010546                  MOV     R5,-(SP)        ;STORE R5 ON THE STACK
051650  010446                  MOV     R4,-(SP)        ;STORE R4 ON THE STACK
051652  013746  177776          MOV     PS,-(SP)        ;STORE PSW ON THE STACK
051656  013737  001452  177776  MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
051664  012004                  MOV     (R0)+,R4        ;STORE QUEUE ADDRESS
051666  011405                  MOV     (R4),R5         ;STORE PARAMETER BLOCK ADDRESS
051670  010566  000010          MOV     R5,10(SP)       ;LOAD ADDRESS ON STACK
051674  001404                  BEQ     1$              ;IF QUEUE IS EMPTY, RETURN
051676  016524  000064          MOV     P.QLNK(R5),(R4)+ ;READJUST HEAD OF THE QUEUE
051702  001001                  BNE     1$              ;IF NOT LAST ELEMENT OF THE QUEUE, RETURN
051704  005014                  CLR     (R4)            ;CLEAR TAIL POINTER IF LAST ELEMENT OF QUEUE

051706  012637  177776  1$:     MOV     (SP)+,PS        ;RESTORE PSW
051712  012604                  MOV     (SP)+,R4        ;RESTORE R4
051714  012605                  MOV     (SP)+,R5        ;RESTORE R5
051716  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.
                                ;*
                                ;;******************  *********************************************

051720  012704  001324          Q.SRCH: MOV     #CINITQ,R4      ;LOAD ADDRESS OF COMMAND INITIATION
                                                                ;  QUEUE IN R4
051724  011405                  1$:     MOV     (R4),R5         ;GET NEXT QUEUE ELEMENT
051726  001425                          BEQ     5$              ;IF END OF QUEUE, RETURN
051730  032765  000002  000014          BIT     #DRVPOS,P.PRST(R5) ;CHECK IF DRIVE IS POSITIONING
051736  001404                          BEQ     2$              ;NO, RETURN PARAMETER BLOCK ADDRESS
051740  010504                          MOV     R5,R4           ;UPDATE LINK ADDRESS POINTER
051742  062704  000064                  ADD     #P.QLNK,R4      ;CALCULATE LINK ADDRESS
051746  000766                          BR      1$              ;GET NEXT ELEMENT OF QUEUE

051750  016514  000064          2$:     MOV     P.QLNK(R5),(R4) ;UPDATE QUEUE LINK
051754  001012                          BNE     5$              ;CHECK IF LAST ELEMENT OF QUEUE
                                                                ;  IF NOT, DO NOT UPDATE QUEUE TAIL
051756  022704  001324                  CMP     #CINITQ,R4      ;CHECK IF ONLY ELEMENT IN QUEUE
051762  001405                          BEQ     4$              ;YES, CLEAR HEAD AND TAIL
051764  162704  000064                  SUB     #P.QLNK,R4      ;ADJUST R4 FOR TAIL POINTER
051770  010437  001326                  MOV     R4,CINITQ+2     ;LOAD TAIL
051774  000207                          RTS     PC              ;RETURN

051776  005024                  4$:     CLR     (R4)+           ;CLEAR HEAD OF QUEUE
052000  005014                          CLR     (R4)            ;CLEAR TAIL OF QUEUE
052002  000207                  5$:     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.
                                        ;*
                                        ;;*****************************************************************

052004  032765  040000  000014  Q.RMOV: BIT     #Q.INIT,P.PRST(R5) ;CHECK IF PARAMETER BLICK ENQUEUED IN
                                                                   ;  INITIATION QUEUE
052012  001001                          BNE     5$                 ;YES, REMOVE PARAMETER BLOCK
                                                                   ;  FROM COMMAND INITIATION QUEUE
052014  000207                          RTS     PC                 ;RETURN

052016  010446                  5$:     MOV     R4,-(SP)           ;STORE R4 ON STACK
052020  010346                          MOV     R3,-(SP)           ;STORE R3 ON THE STACK
052022  012704  001324                  MOV     #CINITQ,R4         ;LOAD ADDRESS OF COMMAND INITIATION
                                                                   ;  QUEUE IN R4
052026  011403                  10$:    MOV     (R4),R3            ;GET NEXT QUEUE ELEMENT
052030  001001                          BNE     11$                ;CHECK IF NO MORE ELEMENTS
052032  000000                          HALT                       ;  *** FLAGS DO NOT AGREE WITH
                                                                   ;  COMMAND INITIATION QUEUE
052034  020305                  11$:    CMP     R3,R5              ;CHECK IF CORRECT PARAMETER BLOCK
052036  001404                          BEQ     12$                ;YES, ADJUST QUEUE
052040  010304                          MOV     R3,R4              ;UPDATE LINK ADDRESS POINTER
052042  062704  000064                  ADD     #P.QLNK,R4         ;CALCULATE LINK ADDRESS
052046  000767                          BR      10$                ;GET NEXT ELEMENT OF QUEUE

052050  016314  000064          12$:    MOV     P.QLNK(R3),(R4)    ;UPDATE QUEUE LINK
052054  001012                          BNE     15$                ;CHECK IF LAST ELEMENT OF QUEUE
                                                                   ;  IF NOT, DO NOT UPDATE QUEUE TAIL
052056  022704  001324                  CMP     #CINITQ,R4         ;CHECK IF ONLY ELEMENT IN QUEUE
052062  001405                          BEQ     14$                ;YES, CLEAR HEAD AND TAIL
052064  162704  000064                  SUB     #P.QLNK,R4         ;ADJUST R4 FOR TAIL POINTER
052070  010437  001326                  MOV     R4,CINITQ+2        ;LOAD TAIL
052074  000402                          BR      15$                ;RETURN

052076  005024                  14$:    CLR     (R4)+              ;CLEAR HEAD OF QUEUE
052100  005014                          CLR     (R4)               ;CLEAR TAIL OF QUEUE
052102  042765  040006  000014  15$:    BIC     #DRVPDT!DRVPOS!Q.INIT,P.PRST(R5) ;CLEAR DRIVE POSITIONING,
                                                                   ;  DRIVE POSITIONING FOR DATA TRANSFER, AND
                                                                   ;  PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE

052110  012603                          MOV     (SP)+,R3           ;RESTORE R3
052112  012604                          MOV     (SP)+,R4           ;RESTORE R4
052114  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
                                ;*
                                ;;*****************************************************************

052116  010546                  C.OPT:  MOV     R5,-(SP)        ;STORE R5 ON STACK
052120  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
052122  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
052124  010246                          MOV     R2,-(SP)        ;STORE R2 ON STACK
052126  010146                          MOV     R1,-(SP)        ;STORE R1 ON STACK
052130  010046                          MOV     R0,-(SP)        ;STORE R0 ON STACK
052132  013746  177776                  MOV     PS,-(SP)        ;STORE PSW ON STACK
052136  012737  000001  001500          MOV     #1,OPTFLG       ;SET FLAG FOR CMND OPTIMIZER
052144  013737  001452  177776          MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
052152  005737  001464                  TST     O.WAIT          ;CHECK IF WAITING FOR COMMAND COMPLETION
052156  001027                          BNE     12$             ;YES, RESTORE REGISTERS
052160  013702  001446                  MOV     RKBAS,R2        ;LOAD R2 WITH BASE ADDRESS OF THE RK06
                                                                ;  REGISTERS
052164  016237  000000  001410          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
052172  032737  000200  001410          BIT     #RDY,T.CS1      ;CHECK IF READY SET
052200  001416                          BEQ     12$             ;NO, WAIT FOR COMMAND COMPLETION
052202  004737  051720                  JSR     PC,Q.SRCH       ;SEARCH FOR PARAMETER BLOCK ADDRESS
052206  005705                          TST     R5              ;CHECK IF NO COMMAND CAN BE ISSUED
052210  001014                          BNE     1$              ;YES, ISSUE COMMAND
052212  032737  040000  001410          BIT     #DI,T.CS1       ;CHECK IF DRIVE INTERRUPT WAITING
052220  001406                          BEQ     12$             ;NO, RETURN
052222  112737  177777  001506          MOVB    #-1,I.ISRL      ;SET FLAG THAT INTERRUPT WAS FORCED
052230  012762  000300  000000          MOV     #INTR,RKCS1(R2) ;FORCE INTERRUPT TO SCAN DRIVE ATTENTIONS
052236  000137  052622          12$:    JMP     O.RTRN          ;RESTORE REGISTERS

052242  112737  000001  001410  1$:     MOVB    #DR.SEL,T.CS1   ;SET DRIVE SELECT BIT
052250  004037  047152                  JSR     R0,I.ISSU       ;ISSUE DRIVE SELECT COMMAND
052254  052622                          O.RTRN                  ;ERROR RETURN ADDRESS
052256  032737  010000  001412          BIT     #NED,T.CS2      ;SEE IF NED ERROR SET
052264  001034                          BNE     24$             ;BR IF YES
052266  032737  000001  001430          BIT     #DRA,T.DS       ;SEE IF DRIVE IS AVAILABLE
052274  001030                          BNE     24$             ;BR IF AVAILABLE
052276  013703  001412                  MOV     T.CS2,R3        ;GET RKCS2
052302  042703  177770                  BIC     #177770,R3      ;CLEAR ALL BUT DRIVE NO.
052306  156337  001514  001512          BISB    INTMSK(R3),W.TIME ;SET BIT FOR THIS DRIVE
052314  006303                          ASL     R3
052316  013763  001476  001544          MOV     W.MIN,W.DRV(R3) ;SET TIMER FOR 1 MINUTE FOR RELEASE
052324  005037  001464                  CLR     O.WAIT          ;CLEAR WAITING FOR CMND COMPLETION
```

```
 9857  052330  004037  047070                 JSR     R0,I.CCLR          ;ISSUE CONTROLLER CLEAR
 9858  052334  052622                         O.RTRN                     ;ERROR RETURN ADDRESS
 9859  052336  042765  040006  000014         BIC     #DRVPOS!DRVPDT!Q.INIT,P.PRST(R5)
 9860  052344  052765  010000  000014  10$:    BIS     #DRVSZD,P.PRST(R5) ;SET DRIVE SIEZED FLAG
 9861  052352  000137  052622                 JMP     O.RTRN             ;EXIT
 9862  052356                          24$:
 9863  052356  005065  000212                 CLR     P.ERR(R5)          ;CLEAR ERROR STATUS INFORMATION
 9864  052362  122765  000164  000001         CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
 9865  052370  001410                         BEQ     2$                 ;YES, CHECK IF IMPLIED SEEK
 9866  052372  122765  000121  000001         CMPB    #RDDATA,P.CMND(R5) ;CHECK IF DATA TRANSFER
 9867  052400  101021                         BHI     3$                 ;NO, GO TO THE COMMAND INITIATOR
 9868  052402  122765  000131  000001         CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
 9869  052410  103415                         BLO     3$                 ;YES, GO TO THE COMMAND INITIATOR
 9870  052412  116504  000000          2$:    MOVB    P.DRVN(R5),R4      ;STORE DRIVE NUMBER
 9871  052416  136437  001514  001513         BITB    INTMSK(R4),O.OVER  ;CHECK IF OVERLAPPED OVERATIONS FOR THIS DRIVE
 9872  052424  001407                         BEQ     3$                 ;NO, GO TO THE COMMAND INITIATOR
 9873  052426  032765  000004  000014         BIT     #DRVPDT,P.PRST(R5) ;CHECK IF POSITIONING FOR
 9874                                                                     ;  DATA TRANSFER OCCURED
 9875  052434  001411                         BEQ     5$                 ;NO, ISSUE SEEK
 9876  052436  042765  040004  000014         BIC     #DRVPDT!Q.INIT,P.PRST(R5) ;CLEAR POSITIONING IN PROGRESS FOR
 9877                                                                     ;  DATA TRANSFER AND PARAMETER BLOCK ENQUEUED
 9878                                                                     ;  IN INITIATION QUEUE
 9879
 9880  052444  010537  052454          3$:    MOV     R5,4$              ;LOAD PARAMETER BLOCK ADDRESS
 9881  052450  004737  050074                 JSR     PC,C.INIT          ;ISSUE COMMAND
 9882  052454  000000                  4$:    .WORD   0                  ;ADDRESS OF PARAMETER BLOCK
 9883                                                                     ;  PLACED HERE
 9884  052456  000461                         BR      O.RTRN             ;RESTORE REGISTERS
 9885
 9886  052460  156437  001514  001512  5$:    BISB    INTMSK(R4),W.TIME  ;SET WATCH DOG TIMING
 9887  052466  010403                         MOV     R4,R3              ;STORE DRIVE NUMBER
 9888  052470  006303                         ASL     R3                 ;MULTIPLY DRIVE NUMBER BY 2
 9889  052472  013763  001472  001544         MOV     W.SEC,W.DRV(R3)    ;LOAD WATCH-DOG TIME
 9890  052500  005037  001464                 CLR     O.WAIT             ;CLEAR WAIT FOR COMMAND
 9891  052504  105037  001506                 CLRB    I.ISRL             ;CLEAR RELEASE OF INTERRUPT ISSUED
 9892
 9893                                  ;      RESET ALL THE BITS IN THE PROGRAM STATUS REGISTEF. EXCEPT
 9894                                  ;              DRIVE IN USE
 9895                                  ;              WRITE FOR WRITE CHECK
 9896                                  ;              NO CHECK
 9897                                  ;              DROP DRIVE FROM TEST SEQUENCE
 9898                                  ;              INHIBIT BUS ADDRESS INCREMENT
 9899  052510  042765  075176  000014         BIC     #^C<DRVUSE!W.WCK!NOCHK!DRPDRV!DTBAII>,P.PRST(R5)
 9900
 9901                                  ;      SET DRIVE POSITIONING, DRIVE POSITIONING BECAUSE OF DATA TRANSFER,
 9902                                  ;        AND PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
 9903  052516  052765  040006  000014         BIS     #DRVPOS!DRVPDT!Q.INIT,P.PRST(R5)
 9904
 9905  052524  010500                         MOV     R5,R0              ;STORE PARAMTER BLOCK ADDRESS
 9906  052526  062700  000016                 ADD     #P.CS1,R0          ;CALCULATE FIRST LOCATION TO BE CLEARED
 9907  052532  010501                         MOV     R5,R1              ;STORE PARAMETER BLOCK ADDRESS
 9908  052534  062701  000062                 ADD     #P.EPAT,R1         ;CALCULATE LAST LOCATION TO BE CLEARED
 9909
 9910  052540  005020                  6$:    CLR     (R0)+              ;CLEAR RETURN PARAMETERS
 9911  052542  020001                         CMP     R0,R1              ;CHECK IF FINISHED
 9912  052544  101775                         BLOS    6$                 ;NO, CLEAR NEXT RETURN PARAMETER
```

```
 9913  052546  005062  000026                    CLR     RKMR1(R2)          ;CLEAR MAINTENANCE REGISTER 1
 9914  052552  016562  000002  000020            MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 9915  052560  010462  000010                    MOV     R4,RKCS2(R2)       ;LOAD DEVICE NUMBER
 9916  052564  116565  000007  000017            MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9917  052572  042765  165777  000016            BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR BITS EXCEPT FORMAT AND
 9918                                                                       ;  DRIVE TYPE
 9919  052600  112765  000117  000016            MOVB    #SEEK,P.CS1(R5) ;STORE COMMAND
 9920  052606  016562  000016  000000            MOV     P.CS1(R5),RKCS1(R2) ;ISSUE SEEK
 9921  052614  004037  051564                    JSR     R0,Q.PUSH          ;REQUEUE COMMAND IN COMMAND
 9922  052620  001324                            CINITQ                     ;  INITIATION QUEUE
 9923
 9924  052622  005037  001500          O.RTRN:   CLR     OPTFLG             ;CLEAR OPTIMIZER FLAG
 9925  052626  012637  177776                    MOV     (SP)+,PS           ;RESTORE PSW
 9926  052632  012600                            MOV     (SP)+,R0           ;RESTORE R0
 9927  052634  012601                            MOV     (SP)+,R1           ;RESTORE R1
 9928  052636  012602                            MOV     (SP)+,R2           ;RESTORE R2
 9929  052640  012603                            MOV     (SP)+,R3           ;RESTORE R3
 9930  052642  012604                            MOV     (SP)+,R4           ;RESTORE R4
 9931  052644  012605                            MOV     (SP)+,R5           ;RESTORE R5
 9932  052646  000207                            RTS     PC                 ;RETURN
```

```
 9933                                   .SBTTL  MEMORY CHECK ENABLE TEST
 9934
 9935  052650  104401  065276           MEMERR: TYPE    ,ERR400          ;TYPE UNEXPECTED MEMORY PARITY TRAP
 9936  052654  011646                           MOV     (SP),-(SP)       ;TYPE PC VALUE
 9937  052656  004737  053224                   JSR     PC,BINOCT
 9938  052662  104401  001165                   TYPE    ,$CRLF
 9939  052666  000002                           RTI                      ;RETURN
```

```
 9940                                   .SBTTL  TYPE ROUTINE
 9941
 9942                                   ;;*****************************************************************
 9943                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 9944                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 9945                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 9946                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 9947                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 9948                                   ;*
 9949                                   ;*CALL:
 9950                                   ;*1) USING A TRAP INSTRUCTION
 9951                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 9952                                   ;*OR
 9953                                   ;*      TYPE
 9954                                   ;*      MESADR
 9955                                   ;*
 9956
 9957  052670  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 9958  052674  100002                          BPL     1$              ;;BR IF YES
 9959  052676  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 9960  052700  000430                          BR      3$              ;;LEAVE
 9961  052702  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 9962  052704  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 9963  052710  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9964  052716  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 9965  052720  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 9966  052726  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 9967  052730  010037  052740                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 9968  052734  004737  055720                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 9969  052740  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
 9970  052742  132737  000040  001211  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 9971  052750  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
 9972  052752  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 9973  052754  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 9974  052756  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 9975  052760  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 9976  052762  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 9977  052766  000002                          RTI                     ;;RETURN
 9978  052770  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 9979  052774  001430                          BEQ     8$
 9980  052776  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 9981  053002  001006                          BNE     5$
 9982  053004  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 9983  053006  104401                          TYPE                    ;;TYPE A CR AND LF
 9984  053010  001165                          $CRLF
 9985  053012  105037  053220                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 9986  053016  000755                          BR      2$              ;;GET NEXT CHARACTER
 9987  053020  004737  053102          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 9988  053024  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 9989  053030  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
 9990  053032  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 9991                                                                  ;;AND THE NULL CHAR.
 9992  053036  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 9993  053042  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 9994  053044  004737  053102                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
 9995  053050  105337  053220                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
 9996  053054  000770                          BR      7$              ;;LOOP
 9997
 9998                                  ;HORIZONTAL TAB PROCESSOR
 9999
10000  053056  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
10001  053062  004737  053102          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
10002  053066  132737  000007  053220          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
10003  053074  001372                          BNE     9$              ;;TAB STOP
10004  053076  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
10005  053100  000724                          BR      2$              ;;GET NEXT CHARACTER
10006  053102                        $TYPEC:
10007  053102  105777  126036                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
10008  053106  100022                          BPL     10$             ;;BR IF NOT                     ;MJD001
10009  053110  017746  126032                  MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
10010  053114  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
10011  053120  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
10012  053124  001012                          BNE     102$            ;;BR IF NOT                     ;MJD001
10013  053126                                                                                          ;MJD001
10014  053126  105777  126012          101$:   TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
10015  053132  100375                          BPL     101$                                            ;MJD001
10016  053134  117716  126006                  MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
10017  053140  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
10018  053144  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
10019  053150  001366                          BNE     101$            ;;BR IF NOT                     ;MJD001
10020  053152                                                                                          ;MJD001
10021  053152  005726                  102$:   TST     (SP)+           ;;FIX STACK                     ;MJD001
10022  053154                                                                                          ;MJD001
10023  053154  105777  125770          10$:    TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
10024  053160  100375                          BPL     10$                                             ;MJD001
10025  053162  116677  000002  125762          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
10026  053170  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
10027  053176  001003                          BNE     1$              ;;BRANCH IF NO
10028  053200  105037  053220                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
10029  053204  000406                          BR      $TYPEX          ;;EXIT
10030  053206  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
10031  053214  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
10032  053216  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
10033  053220  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
10034  053222  000207          $TYPEX: RTS     PC
10035
10036
10037                                  .SBTTL  BINARY TO OCTAL (ASCII) CONVERSION
10038
10039                                  ;;**************************************************************************
10040                                  ;*
10041                                  ;*      THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A
10042                                  ;*      6-DIGIT OCTAL (ASCII) NUMBER.
10043                                  ;*
10044                                  ;*CALL  MOV     NUM,-(SP)
10045                                  ;*      JSR     PC,BINOCT
10046                                  ;*      RETURN                  ;NUMBER IN OCTSTG
10047                                  ;*
10048                                  ;;**************************************************************************
10049
10050  053224                          BINOCT:
10051  053224  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
```

```
10052  053226  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10053  053230  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10054  053232  012704  000006                MOV     #6,R4           ;STORE COUNT
10055  053236  016605  000010                MOV     10(SP),R5       ;PICKUP INPUT NUMBER
10056  053242  012703  053320                MOV     #OCTSTG,R3      ;LOAD ADDRESS OF CONVERTTED STRING
10057  053246  005013                        CLR     (R3)            ;CLEAR OUTPUT BYTE
10058  053250  006105                        ROL     R5              ;ROTATE MSB INTO "C"
10059  053252  000404                        BR      3$              ;GO DO MSB
10060
10061  053254  006105                2$:     ROL     R5              ;FORM THIS DIGIT
10062  053256  006105                        ROL     R5
10063  053260  006105                        ROL     R5
10064  053262  110513                        MOVB    R5,(R3)
10065  053264  106113                3$:     ROLB    (R3)            ;GET LSB OF THIS DIGIT
10066  053266  142713  177770               BICB    #177770,(R3)    ;GET RID OF JUNK
10067  053272  152723  000060               BISB    #'0,(R3)+       ;MAKE IT ASCII
10068  053276  005304                        DEC     R4              ;CHECK IF DONE
10069  053300  001365                        BNE     2$              ;NO CONVERT NEXT DIGIT
10070  053302  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
10071  053304  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
10072  053306  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
10073  053310  012616                        MOV     (SP)+,(SP)      ;SETUP STACK FOR RETURN
10074  053312  104401  053320               TYPE    ,OCTSTG         ;TYPE NUMBER
10075  053316  000207                        RTS     PC              ;RETURN
10076
10077  053320  000006                OCTSTG: .BLKB   6
10078  053326     000     000                .BYTE   0,0
10079
10080                                 .SBTTL  TTY INPUT ROUTINE
10081
10082                                 ;;*****************************************************************
10083                                 .ENABL  LSB
10084
10085                                 .DSABL  LSB
10086
10087
10088                                 ;;*****************************************************************
10089                                 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10090                                 ;*CALL:
10091                                 ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
10092                                 ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10093                                 ;*                              ;;WITH PARITY BIT STRIPPED OFF
10094                                 ;
10095
10096  053330  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
10097  053332  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
10098  053340  105777  125600        1$:     TSTB    @$TKS           ;;WAIT FOR
10099  053344  100375                        BPL     1$              ;;A CHARACTER
10100  053346  117766  125574  000004       MOVB    @$TKB,4(SP)     ;;READ THE TTY
10101  053354  042766  177600  000004       BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
10102  053362  026627  000004  000023       CMP     4(SP),#23       ;;IS IT A CONTROL-S?
10103  053370  001013                        BNE     3$              ;;BRANCH IF NO
10104  053372  105777  125546        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
10105  053376  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
10106  053400  117746  125542               MOVB    @$TKB,-(SP)     ;;GET CHARACTER
10107  053404  042716  177600               BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
```

```
10108  053410  022627  000021          CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
10109  053414  001366                  BNE     2$              ;;IF NOT DISCARD IT
10110  053416  000750                  BR      1$              ;;YES, RESUME
10111  053420  026627  000004  000021  3$:     CMP     4(SP),#$XON     ;;IS IT A RANDOM XON?                   ;RAN001
10112  053426  001744                  BEQ     1$              ;;BRANCH IF YES                         ;RAN001
10113  053430  026627  000004  000140          CMP     4(SP),#140      ;;IS IT UPPER CASE?
10114  053436  002407                  BLT     4$              ;;BRANCH IF YES
10115  053440  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
10116  053446  003003                  BGT     4$              ;;BRANCH IF YES
10117  053450  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
10118  053456  000002                  4$:     RTI             ;;GO BACK TO USER
10119                                  ;;**************************************************************
10120                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10121                                  ;*CALL:
10122                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10123                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10124                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10125
10126  053460  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
10127  053462  005046                  CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
10128  053464  012703  053714          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
10129  053470  022703  053745          2$:     CMP     #$TTYIN+25.,R3  ;;BUFFER FULL?
10130  053474  101456                  BLOS    4$              ;;BR IF YES
10131  053476  104402                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10132  053500  112613                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
10133  053502  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
10134  053506  001022                  BNE     5$              ;;BR IF NO
10135  053510  005716                  TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
10136  053512  001007                  BNE     6$              ;;BR IF NO
10137  053514  112737  000134  053712          MOVB    #'\,9$          ;;TYPE A BACK SLASH
10138  053522  104401  053712                  TYPE    ,9$
10139  053526  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
10140  053532  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
10141  053534  020327  053714                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
10142  053540  103434                  BLO     4$              ;;BR IF YES
10143  053542  111337  053712                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
10144  053546  104401  053712                  TYPE    ,9$             ;;GO TYPE
10145  053552  000746                  BR      2$              ;;GO READ ANOTHER CHAR.
10146  053554  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
10147  053556  001406                  BEQ     7$              ;;BR IF NO
10148  053560  112737  000134  053712          MOVB    #'\,9$          ;;TYPE A BACK SLASH
10149  053566  104401  053712                  TYPE    ,9$
10150  053572  005016                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
10151  053574  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
10152  053600  001003                  BNE     8$              ;;BR IF NO
10153  053602  104401  053745                  TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U'
10154  053606  000726                  BR      1$              ;;GO START OVER
10155  053610  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A '^R'?
10156  053614  001011                  BNE     3$              ;;BRANCH IF NO
10157  053616  105013                  CLRB    (R3)            ;;CLEAR THE CHARACTER
10158  053620  104401  001165                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
10159  053624  104401  053714                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
10160  053630  000717                  BR      2$              ;;GO PICKUP ANOTHER CHACTER
10161  053632  104401  001164          4$:     TYPE    ,$QUES          ;;TYPE A '?'
10162  053636  000712                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
10163  053640  111337  053712          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
```

```
10164  053644  104401  053712                  TYPE    ,9$
10165  053650  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10166  053654  001305                          BNE     2$              ;;LOOP IF NOT RETURN
10167  053656  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10168  053662  104401  001166                  TYPE    ,$LF            ;;TYPE A LINE FEED
10169  053666  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
10170  053670  012603                          MOV     (SP)+,R3        ;;RESTORE R3
10171  053672  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10172  053674  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10173  053702  012766  053714  000004          MOV     #$TTYIN,4(SP)
10174  053710  000002                          RTI                     ;;RETURN
10175  053712     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
10176  053713     000                          .BYTE   0               ;;TERMINATOR
10177  053714  000031                  $TTYIN: .BLKB   25.             ;;RESERVE 25. BYTES FOR TTY INPUT
10178  053745     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
10179  053752  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
10180  053757     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
10181  053764  036440  000040
10182  053770  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
10183  053776  020075     000
10184           054002                  .EVEN
10185
10186                                   .SBTTL  TK INITIALIZE ROUTINE
10187
10188                                   ;;*******************************.*************************************
10189                                   ;*
10190                                   ;*      THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD TO ALL THE
10191                                   ;*        OPERATOR TO COMMUNICATE WITH THE PROGRAM.
10192                                   ;*
10193                                   ;*CALL
10194                                   ;*      JSR     PC,$TKINT
10195                                   ;*      RETURN
10196                                   ;*
10197                                   ;;*****************************************************************
10198
10199  054002  012737  054032  000060  $TKINT: MOV     #$TKSRV,TKVEC   ;INITIALIZE THE KEY BOARD VECTOR
10200  054010  013737  001452  000062          MOV     RKPRI,TKVEC+2   ;LOCK RK06 AND TTY INTERUPTS
10201  054016  005777  125124                  TST     @$TKB           ;CLEAR DONE FLAG
10202  054022  012777  000100  125114          MOV     #IE,@$TKS       ;ENABLE INTERRUPT
10203  054030  000207                          RTS     PC              ;RETURN
10204
10205                                   .SBTTL  TK SERVICE ROUTINE (ONLY CONTROL C IS REGNIZED)
10206
10207  054032  117737  125110  054374  $TKSRV: MOVB    @$TKB,2$        ;PICK UP CHARACTER AND STORE FOR
10208                                                                  ;  PRINT OUT
10209  054040  142737  000200  054374          BICB    #BIT7,2$        ;STRIP THE JUNK
10210  054046  122737  000003  054374          CMPB    #3,2$           ;IS IT A CONTROL-C
10211  054054  001022                          BNE     1$              ;NO, PRINT <CHAR>?<15><12>
10212  054056  104401  054401                  TYPE    ,$CNTLC         ;TYPE A CONTROL-C
10213  054062  005077  125056                  CLR     @$TKS           ;CLEAR INTERRUPT ENABLE
10214  054066  016637  000002  177776          MOV     2(SP),PS        ;REINSTATE PREVIOUS PSW
10215  054074  004737  011656                  JSR     PC,OPRCMD       ;PROCESS OPERATOR COMMAND
10216  054100  013737  001452  177776          MOV     RKPRI,PS        ;LOAD PSW TO LOCK OUT TTY AND RK06
10217  054106  005777  125034                  TST     @$TKB           ;CLEAR DONE FLAG
10218  054112  012777  000100  125024          MOV     #IE,@$TKS       ;ENABLE INTERRUPTS
10219  054120  000002                          RTI                     ;RETURN
```

```
10220
10221  054122  122737  000007  054374  1$:     CMPB    #7,2$           ;CHECK IF CONTROL G
10222  054130  001403                          BEQ     5$              ;YES, GET NEW SWITCH REGISTER VALUE
10223  054132  104401  054374                  TYPE    ,2$             ;PRINT CHARACTER ?<15><12>
10224  054136  000002                  3$:     RTI                     ;RETURN
10225
10226  054140  022737  000176  001140  5$:     CMP     #SWREG,SWR      ;CHECK IF SOFTWARE SWITCH REG
10227  054146  001373                          BNE     3$              ;NO, RETURN
10228  054150  122737  000001  001134          CMPB    #1,$AUTOB       ;CHECK IF IN AUTO ACCEPT MODE
10229  054156  001767                          BEQ     3$              ;YES, RETURN
10230  054160  104401  053752                  TYPE    ,$CNTLG         ;TYPE CONTROL G
10231  054164  104401  053757                  TYPE    ,$MSWR          ;TYPE OLD SWITCH REGISTER
10232  054170  013746  000176                  MOV     $WREG,-(SP)
10233  054174  004737  053224                  JSR     PC,BINOCT
10234  054200  104401  053770                  TYPE    ,$MNEW
10235  054204  005077  124734                  CLR     @$TKS           ;GO INTO FLAG MODE
10236  054210  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
10237  054212  005046                          CLR     -(SP)           ;;THE NEW SWR
10238  054214  105777  124724          7$:     TSTB    @$TKS           ;;CHAR THERE?
10239  054220  100375                          BPL     7$              ;;IF NOT TRY AGAIN
10240
10241  054222  117746  124720                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
10242  054226  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10243
10244  054232  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
10245  054236  001005                          BNE     10$             ;;BRANCH IF NOT
10246  054240  104401  053745                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
10247  054244  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
10248  054250  000757                          BR      19$             ;;LET'S TRY IT AGAIN
10249
10250
10251  054252  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
10252  054256  001016                          BNE     16$             ;;BRANCH IF NO
10253  054260  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
10254  054264  001403                          BEQ     11$             ;;BRANCH IF YES
10255  054266  016677  000002  124644          MOV     2(SP),@SWR      ;;SAVE NEW SWR
10256  054274  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
10257  054300  104401  001165          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
10258  054304  012777  000100  124632          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
10259  054312  000002                  15$:    RTI                     ;;RETURN
10260  054314  004737  053102          16$:    JSR     PC,$TYPEC       ;  ECHO CHAR
10261  054320  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
10262  054324  002420                          BLT     18$             ;;BRANCH IF YES
10263  054326  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
10264  054332  003015                          BGT     18$             ;;BRANCH IF YES
10265  054334  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
10266  054340  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
10267  054344  001403                          BEQ     17$             ;;BRANCH IF YES
10268  054346  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
10269  054350  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
10270  054352  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
10271  054354  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
10272  054360  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
10273  054364  000713                          BR      7$              ;;GET THE NEXT ONE
10274  054366  104401  001164          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
10275  054372  000724                          BR      20$             ;;SIMULATE CONTROL-U
```

```
10276
10277  054374  000                    2$:     .BYTE   0               ;STORAGE FOR QUESTIONABLE INPUT
10278  054375  077  005015     000            .ASCIZ  /?/<15><12>
10279  054401  136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;CONTROL-C
10280
10281                                 .SBTTL  OCTAL TO BINARY CONVERSION ROUTINE
10282
10283                                 ;;*****************************************************************
10284                                 ;*
10285                                 ;*      THIS ROUTINE WILL CHECK A STRING OF ASCII CHARACTERS TERMINATED
10286                                 ;*      WITH A NULL <000> OR COMMA.  IF THE CHARACTERS ARE LEGAL
10287                                 ;*      IT  WILL GENERATE TWO BINARY WORDS PLACING THE LOW 16 BITS
10288                                 ;*      ON THE STACK AND THE HIGH 16 BITS IN LOCATION $HIOCT.
10289                                 ;*
10290                                 ;*CALL
10291                                 ;*      MOV     <ADDRESS OF ASCII STRING>,-(SP)
10292                                 ;*      JSR     PC,OCTBIN
10293                                 ;*      <ADDRESS OF ERROR RETURN>
10294                                 ;*      RETURN
10295                                 ;*
10296                                 ;;*****************************************************************
10297
10298  054406  010046         OCTBIN: MOV     R0,-(SP)        ;SAVE R0
10299  054410  010146                 MOV     R1,-(SP)        ;SAVE R1
10300  054412  010246                 MOV     R2,-(SP)        ;SAVE R2
10301  054414  016600  000010         MOV     10(SP),R0       ;GET ADDRESS OF ASCII STRING
10302  054420  005001                 CLR     R1              ;CLEAR DATA WORDS
10303  054422  005002                 CLR     R2
10304  054424  112046         2$:     MOVB    (R0)+,-(SP)     ;PICK THIS CHARACTER
10305  054426  001423                 BEQ     3$              ;IF ZERO GET OUT
10306  054430  121627  000054         CMPB    (SP),#',        ;CHECK IF COMMA
10307  054434  001420                 BEQ     3$              ;IF COMMA GET OUT
10308  054436  122716  000060         CMPB    #'0,(SP)        ;MAKE SURE THIS CHARACTER IS
10309  054442  003030                 BGT     4$              ;  AN OCTAL DIGIT
10310  054444  122716  000067         CMPB    #'7,(SP)
10311  054450  002425                 BLT     4$
10312  054452  006301                 ASL     R1              ; *2
10313  054454  006102                 ROL     R2
10314  054456  006301                 ASL     R1              ; *4
10315  054460  006102                 ROL     R2
10316  054462  006301                 ASL     R1              ; *8
10317  054464  006102                 ROL     R2
10318  054466  042716  177770         BIC     #^C7,(SP)       ;STRIP THE ASCII JUNK
10319  054472  062601                 ADD     (SP)+,R1        ;ADD THIS DIGIT
10320  054474  000753                 BR      2$              ;LOOP
10321  054476  005726         3$:     TST     (SP)+           ;CLEAN PARTIAL FROM STACK
10322  054500  010166  000010         MOV     R1,10(SP)       ;SAVE RESULT
10323  054504  010237  054540         MOV     R2,$HIOCT
10324  054510  012602                 MOV     (SP)+,R2        ;RESTORE R2
10325  054512  012601                 MOV     (SP)+,R1        ;RESTORE R1
10326  054514  012600                 MOV     (SP)+,R0        ;RESTORE R0
10327  054516  062716  000002         ADD     #2,(SP)         ;ADJUST RETURN
10328  054522  000207                 RTS     PC              ;RETURN
10329
10330  054524  005726         4$:     TST     (SP)+           ;CLEAN UP PARTIAL FROM STACK
10331  054526  012602                 MOV     (SP)+,R2        ;RESTORE R2
```

```
                        MOV     (SP)+,R1        ;RESTORE R1
                        MOV     (SP)+,R0        ;RESTORE R0
                        MOV     @(SP)+,(SP)     ;PUT ADDRESS OF ERROR ROUTINE ON STACK
                        RTS     PC              ;GO PROCESS ERROR
$HIOCT: .WORD   0               ;HIGH ORDER BITS GO HERE
.SBTTL  DECIMAL TO BINARY CONVERSION ROUTINE

;;*************************************************************
;*
;*      THIS ROUTINE WILL CHECK A STRING OF ASCII CHARACTERS TERMINATED
;*      WITH A NULL <000> OR COMMA.  IF THE CHARACTERS ARE LEGAL,
;*      IT WILL GENERATE A  BINARY WORD PLACING IT ON THE STACK.
;*
;*CALL
;*      MOV     <ADDRESS OF ASCII STRING>,-(SP)
;*      JSR     PC,DECBIN
;*      <ADDRESS OF ERROR RETURN>
;*      RETURN
;*
;;*************************************************************

DECBIN: MOV     R0,-(SP)        ;SAVE R0
        MOV     R1,-(SP)        ;SAVE R1
        MOV     R2,-(SP)        ;SAVE R2
        MOV     10(SP),R0       ;GET ADDRESS OF ASCII STRING
        CLR     -(SP)           ;CLEAR DATA WORD
        CLR     R2              ;SIGN SET POSITIVE
        CMPB    #'-,(R0)        ;SEE IF A MINUS SIGN
        BNE     2$              ;BRANCH IF NO MINUS SIGN
        MOVB    (R0)+,R2        ;SAVE FOR LATER USE
2$:     MOVB    (R0)+,R1        ;PICKUP THIS CHARACTER
        BEQ     3$              ;GET OUT IF ZERO
        CMPB    R1,#',          ;CHECK IF COMMA
        BEQ     3$              ;GET OUT IF COMMA
        CMPB    #'0,R1          ;MAKE SURE THIS CHARACTER IS
        BGT     5$              ;  A DIGIT BETWEEN 0 & 9
        CMPB    #'9,R1
        BLT     5$
        BIT     #170000,(SP)    ;DON'T LET NUMBER GET TO BIG
        BNE     5$              ;BRANCH IF NUMBER WOULD OVERFLOW
        ASL     (SP)            ; *2
        MOV     (SP),-(SP)      ;SAVE FOR LATER
        ASL     (SP)            ; *4
        ASL     (SP)            ; *8
        ADD     (SP)+,(SP)      ; *10
        BVS     5$              ;OVERFLOW ISN'T ALLOWED
        SUB     #'0,R1          ;STRIP AWAY THE ASCII JUNK
        ADD     R1,(SP)         ;ADD IN THIS DIGIT
        BVS     5$              ;OVERFLOW ISN'T ALLOWED
        BR      2$              ;LOOP
3$:     TST     R2              ;CHECK IF NUMBER IS NEGATIVE
        BEQ     4$              ;BRANCH IF NO
        NEG     (SP)            ;YES--NEGATE THE NUMBER
4$:     MOV     (SP)+,10(SP)    ;SAVE RESULT
        MOV     (SP)+,R2        ;RESTORE R2
        MOV     (SP)+,R1        ;RESTORE R1
```

```
10388  054670  012600                          MOV     (SP)+,R0        ;RESTORE R0
10389  054672  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN
10390  054676  000207                          RTS     PC              ;RETURN
10391
10392  054700  005726                  5$:     TST     (SP)+           ;CLEAN PARTIAL NUMBER FROM STACK
10393  054702  012602                          MOV     (SP)+,R2        ;RESTORE R2
10394  054704  012601                          MOV     (SP)+,R1        ;RESTORE R1
10395  054706  012600                          MOV     (SP)+,R0        ;RESTORE R0
10396  054710  013616                          MOV     @(SP)+,(SP)     ;PUT ADDRESS OF ERROR ON STACK
10397  054712  000207                          RTS     PC              ;GO PROCESS ERROR
10398                                  .SBTTL  ROUTINE TO SIZE MEMORY
10399
10400                                  ;;**********************************************************************
10401                                  ;*CALL:
10402                                  ;*      JSR     PC,$SIZE
10403                                  ;*      RETURN
10404                                  ;*$LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION
10405
10406  054714  010046                  $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
10407  054716  010146                          MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
10408  054720  013746  000114                  MOV     @#114,-(SP)     ;;SAVE MEMORY ERROR VECTOR PS & PC
10409  054724  013746  000116                  MOV     @#116,-(SP)
10410  054730  012737  000116  000114          MOV     #116,@#114      ;;IGNORE PARITY ERRORS WHILE SIZING
10411  054736  012737  000002  000116          MOV     #RTI,@#116
10412  054744  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
10413  054750  013746  000006                  MOV     @#ERRVEC+2,-(SP)
10414  054754  010600                          MOV     SP,R0           ;;SAVE THE STACK POINTER
10415                                  ;;SET THE ERRVEC PS TO THE PRESENT PS
10416  054756  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
10417  054760  012637  000006                  MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
10418  054764  012737  055004  000004          MOV     #2$,@#ERRVEC    ;;SET FOR TIMEOUT
10419  054772  012701  020000                  MOV     #20000,R1       ;;FIRST ADDRESS
10420  054776  005711                  1$:     TST     (R1)            ;;TEST THIS ADDRESS
10421  055000  005721                          TST     (R1)+           ;;STEP TO NEXT ADDRESS
10422  055002  000775                          BR      1$              ;;TRY ANOTHER
10423  055004  162701  000002          2$:     SUB     #2,R1           ;;DROP BACK
10424  055010  010006                          MOV     R0,SP           ;;RESTORE THE STACK
10425  055012  012637  000006                  MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
10426  055016  012637  000004                  MOV     (SP)+,@#ERRVEC
10427  055022  012637  000116                  MOV     (SP)+,@#116     ;;RESTORE MEMORY ERROR VECTOR
10428  055026  012637  000114                  MOV     (SP)+,@#114
10429  055032  010137  055044                  MOV     R1,$LSTAD       ;;LAST ADDRESS
10430  055036  012601                          MOV     (SP)+,R1        ;;RESTORE R1
10431  055040  012600                          MOV     (SP)+,R0        ;;RESTORE R0
10432  055042  000207                          RTS     PC
10433  055044  000000                  $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
10434                                  ;;**********************************************************************
10435
10436                                  .SBTTL  POWER DOWN AND UP ROUTINES
10437
10438                                  ;POWER DOWN ROUTINE
10439  055046  017737  124066  001404  $PWRDN: MOV     @SWR,SAVSWR     ;SAVE SWITCH REGISTER
10440  055054  012737  055066  000024          MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
10441  055062  000000                          HALT
10442  055064  000776                          BR      .-2             ;HANG UP
10443
```

```
                                        ;POWER UP ROUTINE
055066  005037  055172          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR THE TTY
055072  012737  000144  055174          MOV     #100.,$PWRCT+2
055100  005237  055172          1$:     INC     $PWRCT          ;WAIT FOR THE INCREMENT
055104  001375                          BNE     1$              ;  OF WORD
055106  005337  055174                  DEC     $PWRCT+2
055112  001372                          BNE     1$
055114  012737  055046  000024          MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
055122  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;PRIORITY 7
055130  012737  000340  000036          MOV     #PR7,TRAPVEC+2  ;LOCK OUT ALL INTERRUPTS FOR TRAPS
055136  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
055142  104401  055176                  TYPE    ,$POWER         ;REPORT POWER FAIL
055146  004737  044046                  JSR     PC,PRITIM       ;PRINT TIME
055152  112737  177777  001361          MOVB    #-1,FLAG        ;SET FLAG FOR RESTART SEQUENCE
055160  013777  001404  123752          MOV     SAVSWR,@SWR     ;RESTORE SWITCH REGISTER
055166  000137  007200                  JMP     PWRSRT          ;RESTART PROGRAM

055172  000000  000000          $PWRCT: .WORD   0,0             ;WAIT COUNT FOR TTY
055176  005015  047520  042527  $POWER: .ASCIZ  <15><12>/POWER/<15><12>
055204  006522  000012
                                        .EVEN

                                        .SBTTL  INTEGER MULTIPLY ROUTINE

                                        ;;*******************************************************************
                                        ;*
                                        ;*CALL
                                        ;*      MOV     MULTIPLER,-(SP)
                                        ;*      MOV     MULTIPLICAND,-(SP)
                                        ;*      JSR     PC,$MULT
                                        ;*      RETURN                          ;;PRODUCT IS ON THE STACK
                                        ;*
                                        ;*      STACK   PRODUCT
                                        ;*      -----   -------
                                        ;*
                                        ;*      TOP     LSB'S
                                        ;*      +2      MSB'S
                                        ;*
                                        ;;*******************************************************************

055210                          $MULT:
055210  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
055212  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
055214  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
055216  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
055220  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
055222  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
055224  016605  000016                  MOV     16(SP),R5       ;STORE MULTIPLICAND
055230  016603  000020                  MOV     20(SP),R3       ;STORE MULTIPLIER
055234  005000                          CLR     R0              ;CLEAR HIGH ORDER WORDS
055236  005001                          CLR     R1
055240  005002                          CLR     R2
055242  005004                          CLR     R4
055244  012746  000041                  MOV     #41,-(SP)       ;MOVE 33 DEC TO COUNTER
055250  006000                  1$:     ROR     R0
055252  006001                          ROR     R1
```

```
10500  055254  006002                    ROR     R2              ;SHIFT TO ADD
10501  055256  006003                    ROR     R3
10502  055260  103003                    BCC     2$              ;NO CARRY NO ADD
10503  055262  060501                    ADD     R5,R1
10504  055264  005500                    ADC     R0              ;ADD DOUBLE PRECISION TO OBTAIN NEW PARTIAL
10505  055266  060400                    ADD     R4,R0           ;PRODUCT
10506  055270  005316            2$:     DEC     (SP)            ;DECREMENT COUNTER
10507  055272  001366                    BNE     1$
10508  055274  005726                    TST     (SP)+           ;REMOVE COUNTER
10509  055276  010266  000020           MOV     R2,20(SP)       ;GET LEAST SIGNIFICANT BITS
10510  055302  010366  000016           MOV     R3,16(SP)       ;GET MOST SIGNIFICANT BITS
10511  055306  012605                    MOV     (SP)+,R5        ;;POP STACK INTO R5
10512  055310  012604                    MOV     (SP)+,R4        ;;POP STACK INTO R4
10513  055312  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
10514  055314  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
10515  055316  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
10516  055320  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
10517  055322  000207                    RTS     PC              ;RETURN
10518
10519
10520                          .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
10521
10522                          ;;*****************************************************************
10523                          ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
10524                          ;*WITH A RANGE OF 0 TO 2(+33)-1.
10525                          ;*CALL:
10526                          ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
10527                          ;*      RETURN                  ;;RETURN HERE THE RANDOM
10528                          ;*                              ;;NUMBER WILL BE IN
10529                          ;*                              ;;$HINUM,$LONUM
10530
10531  055324                  $RAND:
10532  055324  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10533  055326  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10534  055330  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10535  055332  013700  055424           MOV     $LONUM,R0       ;;SET R0 WITH LOW
10536  055336  013701  055422           MOV     $HINUM,R1       ;;SET R1 WITH HIGH
10537  055342  012702  177771           MOV     #-7,R2          ;;SET SHIFT COUNT
10538  055346  006300            1$:     ASL     R0              ;;SHIFT R0 LEFT AND
10539  055350  006101                    ROL     R1              ;;ROTATE CARRY INTO R1 AND
10540  055352  005202                    INC     R2              ;;CHECK FOR DONE
10541  055354  001374                    BNE     1$              ;;CONTINUE SHIFT LOOP
10542  055356  063700  055424           ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
10543  055362  005501                    ADC     R1              ;;PROPOGATE CARRY
10544  055364  063701  055422           ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
10545  055370  062700  001057           ADD     #1057,R0        ;;ADD LOW CONSTANT
10546  055374  005501                    ADC     R1              ;;PROPOGATE CARRY
10547  055376  062701  047401           ADD     #47401,R1       ;;ADD HIGH CONSTANT
10548  055402  010037  055424           MOV     R0,$LONUM       ;;SAVE R0
10549  055406  010137  055422           MOV     R1,$HINUM       ;;SAVE R1
10550  055412  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
10551  055414  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
10552  055416  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
10553  055420  000207                    RTS     PC              ;;RETURN
10554  055422  176543            $HINUM: .WORD   176543
10555  055424  123456            $LONUM: .WORD   123456
```

```
10556
10557                                   .SBTTL  SINGLE/DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINES
10558
10559                                   ;;*****************************************************************
10560                                   ;*
10561                                   ;*      THESE ROUTINE WIL CONVERT A 16-BIT OR 32-BIT BINARY NUMBER
10562                                   ;*      TO AN UNSIGNED DECIMAL (ASCII) NUMBER.
10563                                   ;*
10564                                   ;*CALL  MOV     #PNTR,-(SP)     ;POINTER TO LOW WORD OF BINARY NUMBER
10565                                   ;*      JSR     PC,$DB2D
10566                                   ;*      RETURN                  ;DECIMAL NUMBER IN $DECVL
10567                                   ;*
10568                                   ;*CALL  MOV     NUM,-(SP)
10569                                   ;*      JSR     PC,$SB2D
10570                                   ;*      RETURN                  ;DECMAL NUMBER IN $DECVL
10571                                   ;*
10572                                   ;;*****************************************************************
10573
10574  055426                           $SB2D:
10575  055426  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10576  055430  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10577  055432  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10578  055434  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10579  055436  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10580  055440  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10581  055442  016601  000016                   MOV     16(SP),R1       ;STORE NUMBER IN R1
10582  055446  005002                           CLR     R2              ;CLEAR MOST SIGNIFICANT BITS
10583  055450  012737  000005  055574           MOV     #5,$DECNT       ;SET UP FOR 5 CONVERSIONS
10584  055456  012704  055652                   MOV     #$TNPW,R4       ;ADDRESS FOR 5 POWER
10585  055462  012705  055654                   MOV     #$TNPW+2,R5
10586  055466  000421                           BR      $B2D            ;START CONVERSION
10587
10588  055470                           $DB2D:
10589  055470  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10590  055472  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10591  055474  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10592  055476  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10593  055500  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10594  055502  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10595  055504  016602  000016                   MOV     16(SP),R2       ;PICKUP DATA POINTER
10596  055510  012201                           MOV     (R2)+,R1        ;PICKUP BINARY NUMBER
10597  055512  011202                           MOV     (R2),R2
10598  055514  012737  000012  055574           MOV     #10.,$DECNT     ;SET UP TO DO 10 CONVERSIONS
10599  055522  012704  055626                   MOV     #$TNPWR,R4      ;ADDRESS OF TEN POWER
10600  055526  012705  055630                   MOV     #$TNPWR+2,R5
10601
10602  055532  012700  055676           $B2D:   MOV     #$DECVL,R0      ;GET ADDRESS OF "$DECVL" STRING
10603  055536  005003                   $2DEC:  CLR     R3              ;CLEAR PARTIAL
10604  055540  161401                   2$:     SUB     (R4),R1         ;SUBTRACT TEN POWER
10605  055542  005602                           SBC     R2
10606  055544  161502                           SUB     (R5),R2
10607  055546  002402                           BLT     3$              ;BRANCH IF TEN POWER TOO LARGE
10608  055550  005203                           INC     R3              ;ADD 1 TO PARTIAL
10609  055552  000772                           BR      2$              ;LOOP
10610
10611  055554  062401                   3$:     ADD     (R4)+,R1        ;RESTORE SUBTRACTED VALUE
```

```
10612  055556  005502                        ADC     R2
10613  055560  062402                        ADD     (R4)+,R2
10614  055562  022525                        CMP     (R5)+,(R5)+     ;MOVE TO NEXT TEN POWER
10615  055564  052703  000060                BIS     #'0,R3          ;CHANGE IT TO ASCII
10616  055570  110320                        MOVB    R3,(R0)+        ;SAVE ITA
10617  055572  005327                        DEC     (PC)+           ;DONE?
10618  055574  000000              $DECNT:   .WORD   0
10619  055576  001357                        BNE     $2DEC           ;BRANCH IF NO
10620  055600  105020                        CLRB    (R0)+           ;TERMINATOR
10621  055602  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
10622  055604  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
10623  055606  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
10624  055610  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
10625  055612  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
10626  055614  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
10627  055616  012616                        MOV     (SP)+,(SP)      ;SETUP STACK FOR RETURN
10628  055620  104401  055676                TYPE    ,$DECVL
10629  055624  000207                        RTS     PC              ;RETURN
10630
10631  055626  145000              $TNPWR:   145000                  ;1.0E09
10632  055630  035632                        35632
10633  055632  160400                        160400                  ;1.0E08
10634  055634  002765                        2765
10635  055636  113200                        113200                  ;1.0E07
10636  055640  000230                        230
10637  055642  041100                        041100                  ;1.0E06
10638  055644  000017                        17
10639  055646  103240                        103240                  ;1.0E05
10640  055650  000001                        1
10641  055652  023420              $TNPW:    23420                   ;1.0E04
10642  055654  000000                        0
10643  055656  001750                        1750                    ;1.0E03
10644  055660  000000                        0
10645  055662  000144                        144                     ;1.0E02
10646  055664  000000                        0
10647  055666  000012                        12                      ;1.0E01
10648  055670  000000                        0
10649  055672  000001                        1                       ;1.0E00
10650  055674  000000                        0
10651  055676  000014              $DECVL:   .BLKB   12.             ;RESERVE STORAGE FOR ASCIZ STRING
10652                                        .SBTTL  APT COMMUNICATIONS ROUTINE
10653
10654                                        ;;*************************************************************************
10655  055712  112737  000001  056156  $ATY1:   MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
10656  055720  112737  000001  056154  $ATY3:   MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
10657  055726  000403                        BR      $ATYC
10658  055730  112737  000001  056156  $ATY4:   MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
10659  055736                        $ATYC:
10660  055736  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10661  055740  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10662  055742  105737  056154                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
10663  055746  001450                        BEQ     5$              ;;IF NOT:  BR
10664  055750  122737  000001  001210        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
10665  055756  001031                        BNE     3$              ;;IF NOT:  BR
10666  055760  132737  000100  001211        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
10667  055766  001425                        BEQ     3$              ;;IF NOT:  BR
```

```
10668  055770  017600  000004            MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
10669  055774  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10670  056002  005737  001170        1$: TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
10671  056006  001375                    BNE     1$              ;;IF NOT:  WAIT
10672  056010  010037  001204            MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
10673  056014  105720                2$: TSTB    (R0)+           ;;FIND END OF MESSAGE
10674  056016  001376                    BNE     2$
10675  056020  163700  001204            SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
10676  056024  006200                    ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
10677  056026  010037  001206            MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
10678  056032  012737  000004  001170    MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
10679  056040  000413                    BR      5$
10680  056042  017637  000004  056066 3$: MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
10681  056050  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
10682  056056  013746  177776            MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
10683  056062  004737  052670            JSR     PC,$TYPE        ;;CALL TYPE MACRO
10684  056066  000000                4$: .WORD   0
10685  056070                        5$:
10686  056070  105737  056156       10$: TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
10687  056074  001416                    BEQ     12$             ;;IF NOT:  BR
10688  056076  005737  001210            TST     $ENV            ;;RUNNING UNDER APT?
10689  056102  001413                    BEQ     12$             ;;IF NOT:  BR
10690  056104  005737  001170       11$: TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
10691  056110  001375                    BNE     11$             ;;IF NOT:  WAIT
10692  056112  017637  000004  001172    MOV     @4(SP),$FATAL   ;;GET ERROR #
10693  056120  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10694  056126  005237  001170            INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
10695  056132  105037  056156       12$: CLRB    $FFLG           ;;CLEAR FATAL FLAG
10696  056136  105037  056155            CLRB    $LFLG           ;;CLEAR LOG FLAG
10697  056142  105037  056154            CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
10698  056146  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
10699  056150  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
10700  056152  000207                    RTS     PC              ;;RETURN
10701  056154     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
10702  056155     000          $LFLG:  .BYTE   0               ;;LOG FLAG
10703  056156     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
10704          056160                  .EVEN
10705          000200          APTSIZE=200
10706          000001          APTENV=001
10707          000100          APTSPOOL=100
10708          000040          APTCSUP=040
10709                          .SBTTL  TRAP DECODER
10710
10711                          ;;**************************************************************
10712                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10713                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10714                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10715                          ;*GO TO THAT ROUTINE.
10716
10717  056160  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
10718  056162  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
10719  056166  005740                  TST     -(R0)           ;;BACKUP BY 2
10720  056170  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
10721  056172  006300                  ASL     R0              ;;POSITION FOR INDEXING
10722  056174  016000  056214          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
10723  056200  000200                  RTS     R0              ;;GO TO ROUTINE
```

```
10724
10725
10726                                   ;:THIS IS USE TO HANDLE THE "GETPRI" MACRO
10727
10728  056202  011646                   $TRAP2: MOV     (SP),-(SP)      ;:MOVE THE PC DOWN
10729  056204  016666  000004  000002           MOV     4(SP),2(SP)     ;:MOVE THE PSW DOWN
10730  056212  000002                           RTI                     ;:RESTORE THE PSW
10731
10732                                   .SBTTL  TRAP TABLE
10733
10734                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10735                                   ;*BY THE "TRAP" INSTRUCTION.
10736
10737                                   ;       ROUTINE
10738                                   ;       -------
10739  056214  056202                   $TRPAD: .WORD   $TRAP2
10740  056216  052670                           $TYPE   ;:CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
10741
10742
10743  056220  053330                           $RDCHR  ;:CALL=RDCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
10744  056222  053460                           $RDLIN  ;:CALL=RDLIN    TRAP+3(104403)  TTY TYPEIN STRING ROUTINE
```

```
                                        .SBTTL  ASCII MESSAGES

                                XXDPMG: .ASCII  <CR><LF>/TO TEST DRIVE 0,REMOVE XXDP MEDIA,/
                                        .ASCII  <CR><LF>/CLEAR LOC 40,AND HIT CONT./

                                        .ASCIZ  <CR><LF>/IF DRIVE 0 ISN'T TO BE TESTED,JUST HIT CONT. ./<CR><LF>

                                ILLCOM: .ASCII  /ILLEGAL OPERATOR COMMAND /

                                ILLCMD: .ASCIZ  /  ?/<15><12>
                                OPR000: .ASCII  /DRIVE NUMBER /

                                OPR001: .ASCIZ  /0 /
                                OPR002: .ASCIZ  /ILLEGAL?/<15><12>

                                OPR003: .ASCIZ  /UNDER TEST/

                                OPR004: .ASCIZ  /ALREADY ASSIGNED?/<15><12>

                                OPR005: .ASCIZ  /NOT ASSIGNED?/<15><12>

                                OPR006: .ASCIZ  /HAS BEEN DROPPED FROM TEST SEQUENCE/<15><12>

                                OPR007: .ASCIZ  /PLEASE ENTER COMMAND/<15><12>/*/

                                OPR008: .ASCIZ  /OPERATOR INITIATED DROP DRIVE DURING WRITE PACK/
```

```
10801  056672  044524  052101  042105
10802  056700  042040  047522  020120
10803  056706  051104  053111  020105
10804  056714  052504  044522  043516
10805  056722  053440  044522  042524
10806  056730  050040  041501  000113
10807  056736  005015  051104  053111  OPR010: .ASCII  <15><12>/DRIVE NUMBER /
10808  056744  020105  052516  041115
10809  056752  051105     040
10810  056755     060  000040          OPR011: .ASCIZ  /0 /
10811  056760  051104  053111  020105  OPR012: .ASCIZ  /DRIVE WRITE LOCKED/<15><12>
10812  056766  051127  052111  020105
10813  056774  047514  045503  042105
10814  057002  005015     000
10815  057005     101  046114  042040  OPR013: .ASCIZ  /ALL DRIVES CURRENTLY UNDER TEST/<15><12>
10816  057012  044522  042526  020123
10817  057020  052503  051122  047105
10818  057026  046124  020131  047125
10819  057034  042504  020122  042524
10820  057042  052123  005015     000
10821  057047     116  020117  051104  OPR014: .ASCIZ  /NO DRIVE IN USE/<15><12>
10822  057054  053111  020105  047111
10823  057062  052440  042523  005015
10824  057070     000
10825  057071     101  046114  043511  OPR015: .ASCIZ  /ALLIGNMENT PACK IN DRIVE/<15><12>
10826  057076  046516  047105  020124
10827  057104  040520  045503  044440
10828  057112  020116  051104  053111
10829  057120  006505  000012
10830  057124  025052  051104  053111  DRVSTT: .ASCIZ  /**DRIVE STATISTICS**/<15><12>
10831  057132  020105  052123  052101
10832  057140  051511  044524  051503
10833  057146  025052  005015     000
10834  057153     015  042012  044522  STT000: .ASCII  <15><12>/DRIVE SERIAL NO. /
10835  057160  042526  051440  051105
10836  057166  040511  020114  047516
10837  057174  020056
10838  057176  000003                  STT100: .BLKB   3
10839  057201     000                          .BYTE   0
10840  057202  051117  042504  051522  STT001: .ASCIZ  /ORDERS PERFORMED /
10841  057210  050040  051105  047506
10842  057216  046522  042105  000040
10843  057224  005015  047527  042122  STT002: .ASCIZ  <15><12>/WORDS WRITTEN*65K /
10844  057232  020123  051127  052111
10845  057240  042524  025116  032466
10846  057246  020113     000
10847  057251     015  053412  051117  STT003: .ASCIZ  <15><12>/WORDS READ*65K /
10848  057256  051504  051040  040505
10849  057264  025104  032466  020113
10850  057272     000
10851  057273     015  051412  043117  STT004: .ASCIZ  <15><12>/SOFT DATA ERRORS/<15><12>/  ECC /
10852  057300  020124  040504  040524
10853  057306  042440  051122  051117
10854  057314  006523  020012  042440
10855  057322  041503  000040
10856  057326  005015  020040  042522  STT005: .ASCIZ  <15><12>/  REREAD /
```

```
10857  057334  042522  042101  000040
10858  057342  005015  020040  043117  STT006: .ASCIZ  <15><12>/  OFFSET /
10859  057350  051506  052105  000040
10860  057356  005015  040510  042122  STT007: .ASCIZ  <15><12>/HARD DATA ERRORS /
10861  057364  042040  052101  020101
10862  057372  051105  047522  051522
10863  057400  000040
10864  057402  005015  042523  045505  STT008: .ASCIZ  <15><12>/SEEK INCOMPLETES /
10865  057410  044440  041516  046517
10866  057416  046120  052105  051505
10867  057424  000040
10868  057426  005015  050117  051105  STT009: .ASCIZ  <15><12>"OPERATION INCOMPLETES/MISPOSITIONS "
10869  057434  052101  047511  020116
10870  057442  047111  047503  050115
10871  057450  042514  042524  027523
10872  057456  044515  050123  051517
10873  057464  052111  047511  051516
10874  057472  000040
10875  057474  005015  052117  042510  STT010: .ASCIZ  <15><12>/OTHER ERRORS /
10876  057502  020122  051105  047522
10877  057510  051522  000040
10878  057514  005015  040520  045503  STT011: .ASCII  <15><12>/PACK SERIAL NUMBER /
10879  057522  051440  051105  040511
10880  057530  020114  052516  041115
10881  057536  051105     040
10882  057541  000013            STT012: .BLKB   11.
10883  057554  005015     000    STT013: .ASCIZ  <15><12>
10884  057557     103  052520  044440  SYS000: .ASCIZ  /CPU ID=/
10885  057564  036504     000
10886  057567     117  042526  046122  SYS001: .ASCIZ  /OVERLAY LOADER=/
10887  057574  054501  046040  040517
10888  057602  042504  036522     000
10889  057607     115  054101  052040  SYS002: .ASCIZ  /MAX TRANSFER SIZE ( MAX: /
10890  057614  040522  051516  042506
10891  057622  020122  044523  042532
10892  057630  024040  046440  054101
10893  057636  020072     000
10894  057641     040  036451     000  SYS003: .ASCIZ  / )=/
10895  057645     116  027117  047440  SYS004: .ASCIZ  /NO. OF SOFTWARE COMPARES=/
10896  057652  020106  047523  052106
10897  057660  040527  042522  041440
10898  057666  046517  040520  042522
10899  057674  036523     000
10900  057677     122  030113  026466  SYS005: .ASCIZ  /RK06-RK07 BUS ADD=/
10901  057704  045522  033460  041040
10902  057712  051525  040440  042104
10903  057720  000075
10904  057722  045522  033060  051055  SYS006: .ASCIZ  /RK06-RK07 VEC ADD=/
10905  057730  030113  020067  042526
10906  057736  020103  042101  036504
10907  057744     000
10908  057745     116  047117  054105  SYS007: .ASCIZ  /NONEXISTENT MEMORY/<15><12>
10909  057752  051511  042524  052116
10910  057760  046440  046505  051117
10911  057766  006531  000012
10912  057772  042522  042101  020131  SYS008: .ASCIZ  /READY TO BEGIN PERFORMANCE TESTING/<15><12><12>
```

```
10913  060000  047524  041040  043505
10914  060006  047111  050040  051105
10915  060014  047506  046522  047101
10916  060022  042503  052040  051505
10917  060030  044524  043516  005015
10918  060036  000012
10919  060040  052123  052101  051511  SYS009: .ASCIZ  /STATISTIC INTERVAL=/
10920  060046  044524  020103  047111
10921  060054  042524  053122  046101
10922  060062  000075
10923  060064  045522  033060  051055  SYS010: .ASCIZ  /RK06-RK07 PRIOR=/
10924  060072  030113  020067  051120
10925  060100  047511  036522     000
10926  060105     120  051101  042515  PAR000: .ASCII  /PARMETERS FOR DRIVE /
10927  060112  042524  051522  043040
10928  060120  051117  042040  044522
10929  060126  042526     040
10930  060131     060  005015  000012  PAR001: .ASCIZ  /0/<15><12><12>
10931  060136  054503  044514  042116  PAR002: .ASCIZ  /CYLINDER MAX,MIN =/
10932  060144  051105  046440  054101
10933  060152  046454  047111  036440
10934  060160     000
10935  060161     124  040522  045503  PAR003: .ASCIZ  /TRACK MAX,MIN =/
10936  060166  046440  054101  046454
10937  060174  047111  036440     000
10938  060201     123  041505  047524  PAR004: .ASCIZ  /SECTOR MAX,MIN =/
10939  060206  020122  040515  026130
10940  060214  044515  020116  000075
10941  060222  042522  042101  053457  PAR005: .ASCIZ  'READ/WRITE RATIO ='
10942  060230  044522  042524  051040
10943  060236  052101  047511  036440
10944  060244     000
10945  060245     127  044522  042524  PAR006: .ASCIZ  /WRITE CHECK AFTER WRITE =/
10946  060252  041440  042510  045503
10947  060260  040440  052106  051105
10948  060266  053440  044522  042524
10949  060274  036440     000
10950  060277     103  051117  042522  PAR007: .ASCIZ  /CORRECTABLE READ ERROR THRESHOLD =/
10951  060304  052103  041101  042514
10952  060312  051040  040505  020104
10953  060320  051105  047522  020122
10954  060326  044124  042522  044123
10955  060334  046117  020104  000075
10956  060342  047125  047503  051122  PAR008: .ASCIZ  /UNCORRECTABLE READ ERROR THRESHOLD =/
10957  060350  041505  040524  046102
10958  060356  020105  042522  042101
10959  060364  042440  051122  051117
10960  060372  052040  051110  051505
10961  060400  047510  042114  036440
10962  060406     000
10963  060407     123  042505  020113  PAR009: .ASCIZ  /SEEK ERROR THRESHOLD =/
10964  060414  051105  047522  020122
10965  060422  044124  042522  044123
10966  060430  046117  020104  000075
10967  060436  047111  044510  044502  PAR010: .ASCIZ  /INHIBIT ERROR CORRECTION AND RETRY =/
10968  060444  020124  051105  047522
```

```
10969  060452  020122  047503  051122
10970  060460  041505  044524  047117
10971  060466  040440  042116  051040
10972  060474  052105  054522  036440
10973  060502     000
10974  060503     117  042520  040522  PAR011: .ASCIZ  /OPERATION COUNT THRESHOLD*65K =/
10975  060510  044524  047117  041440
10976  060516  052517  052116  052040
10977  060524  051110  051505  047510
10978  060532  042114  033052  045465
10979  060540  036440     000
10980  060543     127  051117  051504  PAR012: .ASCIZ  /WORDS TRANSFERRED THRESHOLD*1,049K =/
10981  060550  052040  040522  051516
10982  060556  042506  051122  042105
10983  060564  052040  051110  051505
10984  060572  047510  042114  030452
10985  060600  030054  034464  020113
10986  060606  000075
10987  060610  044506  051522  000124  PAR013: .ASCIZ  /FIRST/
10988  060616  042523  047503  042116  PAR014: .ASCIZ  /SECOND/
10989  060624     000
10990  060625     124  044510  042122  PAR015: .ASCIZ  /THIRD/
10991  060632     000
10992  060633     106  052517  052122  PAR016: .ASCIZ  /FOURTH/
10993  060640  000110
10994  060642  044506  052106  000110  PAR017: .ASCIZ  /FIFTH/
10995  060650  042440  041530  052514  PAR018: .ASCIZ  / EXCLUDED PACK AREA =/
10996  060656  042504  020104  040520
10997  060664  045503  040440  042522
10998  060672  020101  000075
10999  060676  044103  047101  042507  PAR019: .ASCIZ  /CHANGE/
11000  060704     000
11001  060705     125  040516  046102  PAR020: .ASCII  /UNABLE TO GENERATE NEW COMMAND DUE TO/<15><12>
11002  060712  020105  047524  043440
11003  060720  047105  051105  052101
11004  060726  020105  042516  020127
11005  060734  047503  046515  047101
11006  060742  020104  052504  020105
11007  060750  047524  005015
11008  060754  054105  046103  042125          .ASCIZ  /EXCLUDED PACK AREAS AND MAX TRANSFER SIZE CONFLICT/<15><12>
11009  060762  042105  050040  041501
11010  060770  020113  051101  040505
11011  060776  020123  047101  020104
11012  061004  040515  020130  051124
11013  061012  047101  043123  051105
11014  061020  051440  055111  020105
11015  061026  047503  043116  044514
11016  061034  052103  005015     000
11017  061041     123  046501  046120  PAR021: .ASCIZ  /SAMPLED COMPARES =/
11018  061046  042105  041440  046517
11019  061054  040520  042522  020123
11020  061062  000075
11021  061064  042524  052123  047111  PAR022: .ASCIZ  /TESTING RK06? Y OR N  /
11022  061072  020107  045522  033060
11023  061100  020077  020131  051117
11024  061106  047040  020040     000
```

```
11025  061113     040  037440  037477  QUESMK: .ASCIZ  /  ??????  /
11026  061120  037477  020077  000040
11027  061126  020040     000          BLANKS: .ASCIZ  /  /
11028  061131     052  025052  040506  ERR000: .ASCIZ  /***FATAL ERROR***/<15><12><12>
11029  061136  040524  020114  051105
11030  061144  047522  025122  025052
11031  061152  005015  000012
11032  061156  047503  052116  047522  ERR001: .ASCIZ  /CONTROLLER CLEAR DID NOT CLEAR ERROR/
11033  061164  046114  051105  041440
11034  061172  042514  051101  042040
11035  061200  042111  047040  052117
11036  061206  041440  042514  051101
11037  061214  042440  051122  051117
11038  061222     000
11039  061223     116  020117  052101  ERR002: .ASCIZ  /NO ATTENTION IN RKASOF/
11040  061230  042524  052116  047511
11041  061236  020116  047111  051040
11042  061244  040513  047523  000106
11043  061252  047125  054105  042520  ERR003: .ASCIZ  /UNEXPECTED DATA TYPE ERROR/
11044  061260  052103  042105  042040
11045  061266  052101  020101  054524
11046  061274  042520  042440  051122
11047  061302  051117     000
11048  061305     104  044522  042526  ERR004: .ASCIZ  /DRIVE CLEAR DID NOT RESET ATTENTION/
11049  061312  041440  042514  051101
11050  061320  042040  042111  047040
11051  061326  052117  051040  051505
11052  061334  052105  040440  052124
11053  061342  047105  044524  047117
11054  061350     000
11055  061351     111  046114  043505  ERR005: .ASCIZ  /ILLEGAL DRIVER COMMAND/
11056  061356  046101  042040  044522
11057  061364  042526  020122  047503
11058  061372  046515  047101  000104
11059  061400  052515  052114  050111  ERR006: .ASCIZ  /MULTIPLE DRIVE SELECT/
11060  061406  042514  042040  044522
11061  061414  042526  051440  046105
11062  061422  041505  000124
11063  061426  042523  020124  053523  ERR007: .ASCIZ  /SET SWITCH 14 TO CYCLE ON ERROR/<15><12>
11064  061434  052111  044103  030440
11065  061442  020064  047524  041440
11066  061450  041531  042514  047440
11067  061456  020116  051105  047522
11068  061464  006522  000012
11069  061470  047503  052116  030122  ERR008: .ASCIZ  /CONTROLLER ERROR THRESHOLD EXCEEDED/
11070  061476  046114  051105  042440
11071  061504  051122  051117  052040
11072  061512  051110  051505  047510
11073  061520  042114  042440  041530
11074  061526  042505  042504  000104
11075  061534  040504  040524  046040  ERR009: .ASCIZ  /DATA LATE THRESHOLD EXCEEDED/
11076  061542  052101  020105  044124
11077  061550  042522  044123  046117
11078  061556  020104  054105  042503
11079  061564  042105  042105     000
11080  061571     101  042104  042522  ERR012: .ASCII  /ADDRESS GOOD    BAD     SECTOR  DATA/<15><12>
```

```
 11081  061576  051523  043440  047517
 11082  061604  020104  020040  041040
 11083  061612  042101  020040  020040
 11084  061620  051440  041505  047524
 11085  061626  020122  042040  052101
 11086  061634  006501     012
 11087  061637     040  020040  020040          .ASCIZ  /       DATA    DATA    POS     PAT #/<15><12>
 11088  061644  020040  042040  052101
 11089  061652  020101  020040  042040
 11090  061660  052101  020101  020040
 11091  061666  050040  051517  020040
 11092  061674  020040  050040  052101
 11093  061702  021440  005015     000
 11094  061707     103  047117  051124  ERR013: .ASCIZ  /CONTROLLER ERROR NOT FLAGGED/<15><12>
 11095  061714  046117  042514  020122
 11096  061722  051105  047522  020122
 11097  061730  047516  020124  046106
 11098  061736  043501  042507  006504
 11099  061744  000012
 11100  061746  047527  042122  041440  ERR014: .ASCIZ  /WORD COUNT NOT EQUAL 0/<15><12>
 11101  061754  052517  052116  047040
 11102  061762  052117  042440  052521
 11103  061770  046101  030040  005015
 11104  061776     000
 11105  061777     102  051525  040440  ERR015: .ASCIZ  /BUS ADD INCORRECT/<15><12>
 11106  062004  042104  044440  041516
 11107  062012  051117  042522  052103
 11108  062020  005015     000
 11109  062023     103  046131  052054  ERR016: .ASCIZ  /CYL,TRK,SEC INCORRECT/<15><12>
 11110  062030  045522  051454  041505
 11111  062036  044440  041516  051117
 11112  062044  042522  052103  005015
 11113  062052     000
 11114  062053     103  047117  051124  ERR017: .ASCIZ  /CONTROLLER COMMAND TIME OUT/
 11115  062060  046117  042514  020122
 11116  062066  047503  046515  047101
 11117  062074  020104  044524  042515
 11118  062102  047440  052125     000
 11119  062107     120  047522  020107  ERR020: .ASCIZ  /PROG ERROR/
 11120  062114  051105  047522  000122
 11121  062122  046111  020114  052506  ERR021: .ASCIZ  /ILL FUNCT CODE/
 11122  062130  041516  020124  047503
 11123  062136  042504     000
 11124  062141     106                  ERR022: .BYTE   106
 11125  062142  051105  047522  020122          .ASCIZ  /ERROR WHILE WAITING TO REPORT ERROR/
 11126  062150  044127  046111  020105
 11127  062156  040527  052111  047111
 11128  062164  020107  047524  051040
 11129  062172  050105  051117  020124
 11130  062200  051105  047522  000122
 11131  062206     106                  ERR023: .BYTE   106
 11132  062207     116  047117  042455          .ASCIZ  /NON-EXIST DRV/
 11133  062214  044530  052123  042040
 11134  062222  053122     000
 11135  062225     106                  ERR024: .BYTE   106
 11136  062226  042523  041522  047117          .ASCIZ  /SERCON PAR/
```

```
11137  062234  050040  051101     000
11138  062241     106                    ERR025: .BYTE   106
11139  062242  047125  052111  043040             .ASCIZ  /UNIT FIELD ERR/
11140  062250  042511  042114  042440
11141  062256  051122     000
11142  062261     015  044412  052116    ERR026: .ASCIZ  <15><12>/INT FROM DRV NOT UNDER TEST/<15><12>
11143  062266  043040  047522  020115
11144  062274  051104  020126  047516
11145  062302  020124  047125  042504
11146  062310  020122  042524  052123
11147  062316  005015     000
11148  062321     110  051101  020104    ERR027: .ASCIZ  /HARD DRV ERR/<15><12>
11149  062326  051104  020126  051105
11150  062334  006522  000012
11151  062340  051104  020126  052123    ERR028: .ASCIZ  /DRV STATUS CHANGE DID NOT CLR/<15><12>
11152  062346  052101  051525  041440
11153  062354  040510  043516  020105
11154  062362  044504  020104  047516
11155  062370  020124  046103  006522
11156  062376  000012
11157  062400     106                    ERR029: .BYTE   106
11158  062401     104  044522  042526             .ASCIZ  /DRIVE BECAME NOT AVAIL/
11159  062406  041040  041505  046501
11160  062414  020105  047516  020124
11161  062422  053101  044501  000114
11162  062430  047125  054105  042520    ERR030: .ASCIZ  /UNEXPECT ATTN/<15><12>
11163  062436  052103  040440  052124
11164  062444  006516  000012
11165  062450  046111  020114  051504    ERR031: .ASCIZ  /ILL DSK ADD/
11166  062456  020113  042101  000104
11167  062464  051127  020124  047514    ERR032: .ASCIZ  /WRT LOCK ERR/
11168  062472  045503  042440  051122
11169  062500     000
11170  062501     101  020103  047514    ERR033: .ASCIZ  /AC LOW/
11171  062506  000127
11172  062510  050123  042505  020104    ERR034: .ASCIZ  /SPEED LOSS/
11173  062516  047514  051523     000
11174  062523     104  053122  052040    ERR035: .ASCIZ  /DRV TYPE ERR/
11175  062530  050131  020105  051105
11176  062536  000122
11177  062540  047506  046522  052101    ERR036: .ASCIZ  /FORMAT ERR/
11178  062546  042440  051122     000
11179  062553     116  047117  042455    ERR037: .ASCIZ  /NON-EXIST DRV FUNCT/
11180  062560  044530  052123  042040
11181  062566  053122  043040  047125
11182  062574  052103     000
11183  062577     106                    ERR038: .BYTE   106
11184  062600  042523  045505  044440             .ASCIZ  /SEEK INCOMP/
11185  062606  041516  046517  000120
11186  062614     106                    ERR039: .BYTE   106
11187  062615     104  053122  047440             .ASCIZ  /DRV OFF TRK/
11188  062622  043106  052040  045522
11189  062630     000
11190  062631     106                    ERR040: .BYTE   106
11191  062632  044515  050123  051517             .ASCIZ  /MISPOS/
11192  062640     000
```

```
11193  062641     126                        ERR042: .BYTE   126
11194  062642  050117  051105  052101                .ASCIZ  /OPERAT INCOMP/
11195  062650  044440  041516  046517
11196  062656  000120
11197  062660     127                        ERR043: .BYTE   127
11198  062661     110  051104  053040                .ASCIZ  /HDR VRC ERR/
11199  062666  041522  042440  051122
11200  062674     000
11201  062675     116  047117  042455  ERR044: .ASCIZ  /NON-EXIST MEM/
11202  062702  044530  052123  046440
11203  062710  046505     000
11204  062713     125  044516  052502  ERR045: .ASCIZ  /UNIBUS PAR/
11205  062720  020123  040520  000122
11206  062726     117                        ERR046: .BYTE   117
11207  062727     104  053122  052040                .ASCIZ  /DRV TIMING ERR/
11208  062734  046511  047111  020107
11209  062742  051105  000122
11210  062746  040504  040524  046040  ERR047: .ASCIZ  /DATA LATE/<15><12>
11211  062754  052101  006505  000012
11212  062762     117                        ERR048: .BYTE   117
11213  062763     104  052101  020101                .ASCIZ  /DATA CHK/
11214  062770  044103  000113
11215  062774     107                        ERR050: .BYTE   107
11216  062775     127  044522  042524                .ASCIZ  /WRITE CHK ERR/
11217  063002  041440  045510  042440
11218  063010  051122     000
11219  063013     103  046131  040440  ERR051: .ASCIZ  /CYL ADDR OVRFLW/
11220  063020  042104  020122  053117
11221  063026  043122  053514     000
11222  063033     106                        ERR052: .BYTE   106
11223  063034  046503  042116  052040                .ASCIZ  /CMND TIME OUT DURING POSITIONING OPERATION/<15><12>
11224  063042  046511  020105  052517
11225  063050  020124  052504  044522
11226  063056  043516  050040  051517
11227  063064  052111  047511  044516
11228  063072  043516  047440  042520
11229  063100  040522  044524  047117
11230  063106  005015     000
11231  063111     106                        ERR053: .BYTE   106
11232  063112  051104  020126  047516                .ASCIZ  /DRV NOT RDY AFTER START SPINDLE/
11233  063120  020124  042122  020131
11234  063126  043101  042524  020122
11235  063134  052123  051101  020124
11236  063142  050123  047111  046104
11237  063150  000105
11238  063152     106                        ERR054: .BYTE   106
11239  063153     126  046117  053040                .ASCIZ  /VOL VALID DID NOT SET AFTER PACK ACK/
11240  063160  046101  042111  042040
11241  063166  042111  047040  052117
11242  063174  051440  052105  040440
11243  063202  052106  051105  050040
11244  063210  041501  020113  041501
11245  063216  000113
11246  063220  051104  020126  047125  ERR056: .ASCIZ  /DRV UNSAFE/
11247  063226  040523  042506     000
11248  063233     104  053122  042440  ERR057: .ASCIZ  /DRV ERR NOT INDICATED BY FAULT/<15><12>
```

```
11249  063240  051122  047040  052117
11250  063246  044440  042116  041511
11251  063254  052101  042105  041040
11252  063262  020131  040506  046125
11253  063270  006524  000012
11254  063274  051104  020126  052101  ERR058: .ASCIZ  /DRV ATTN BUT NO FAULT OR DRV STATUS CHANGE/<15><12>
11255  063302  047124  041040  052125
11256  063310  047040  020117  040506
11257  063316  046125  020124  051117
11258  063324  042040  053122  051440
11259  063332  040524  052524  020123
11260  063340  044103  047101  042507
11261  063346  005015     000
11262  063351     103  047115  020104  ERR059: .ASCIZ  /CMND TIME OUT DRV SEIZED BY OTHER PORT/<15><12>
11263  063356  044524  042515  047440
11264  063364  052125  042040  053122
11265  063372  051440  044505  042532
11266  063400  020104  054502  047440
11267  063406  044124  051105  050040
11268  063414  051117  006524  000012
11269  063422  041505  020103  044515  ERR060: .ASCIZ  /ECC MISCORRECTION/<15><12>
11270  063430  041523  051117  042522
11271  063436  052103  047511  006516
11272  063444  000012
11273  063446  040504  040524  041440  ERR062: .ASCIZ  /DATA COMPARE ERROR/<15><12>
11274  063454  046517  040520  042522
11275  063462  042440  051122  051117
11276  063470  005015     000
11277  063473     126                  ERR064: .BYTE   126
11278  063474  044515  050123  051517          .ASCIZ  /MISPOS/
11279  063502     000
11280  063503     106                  ERR065: .BYTE   106
11281  063504  042510  042101  051440          .ASCIZ  /HEAD SELECT ERROR/
11282  063512  046105  041505  020124
11283  063520  051105  047522  000122
11284  063526  041505  020103  047514  ERR066: .ASCIZ  /ECC LOGIC ERROR/
11285  063534  044507  020103  051105
11286  063542  047522  000122
11287  063546  045522  041505  052120  ERR067: .ASCIZ  /RKECP1  RKECPS/<15><12>
11288  063554  020040  045522  041505
11289  063562  051520  005015     000
11290  063567     127  051117  020104  ERR070: .ASCIZ  /WORD COUNT INVALID/
11291  063574  047503  047125  020124
11292  063602  047111  040526  044514
11293  063610  000104
11294  063612  052502  020123  042101  ERR071: .ASCIZ  /BUS ADDRESS INVALID/
11295  063620  051104  051505  020123
11296  063626  047111  040526  044514
11297  063634  000104
11298  063636  054503  044514  042116  ERR072: .ASCIZ  /CYLINDER INVALID/
11299  063644  051105  044440  053116
11300  063652  046101  042111     000
11301  063657     124  040522  045503  ERR073: .ASCIZ  ''TRACK/SECTOR INVALID''
11302  063664  051457  041505  047524
11303  063672  020122  047111  040526
11304  063700  044514  000104
```

```
11305  063704     106                    ERR074: .BYTE   106
11306  063705     105  051122  051117            .ASCIZ  /ERROR IN DRV DID NOT RELOAD HEADS/
11307  063712  044440  020116  051104
11308  063720  020126  044504  020104
11309  063726  047516  020124  042522
11310  063734  047514  042101  044040
11311  063742  040505  051504     000
11312  063747     103  051125  042522  ERR100: .ASCII  /CURRENT SUPPLIED/<15><12>
11313  063754  052116  051440  050125
11314  063762  046120  042511  006504
11315  063770     012
11316  063771     103  047115  020104          .ASCIZ  /CMND    CYL     TRK     SEC     OFFSET  BUS AD  WRD CT/
11317  063776  020040  041440  046131
11318  064004  020040  020040  052040
11319  064012  045522  020040  020040
11320  064020  051440  041505  020040
11321  064026  020040  047440  043106
11322  064034  042523  020124  041040
11323  064042  051525  040440  020104
11324  064050  053440  042122  041440
11325  064056  000124
11326  064060  051120  053105  047511  ERR101: .ASCII  /PREVIOUS /<15><12>
11327  064066  051525  006440     012
11328  064073     103  047115  020104          .ASCIZ  /CMND    CYL     TRK     SEC     WRD CT/
11329  064100  020040  041440  046131
11330  064106  020040  020040  052040
11331  064114  045522  020040  020040
11332  064122  051440  041505  020040
11333  064130  020040  053440  042122
11334  064136  041440  000124
11335  064142  020040  040520  000124  ERR102: .ASCIZ  /  PAT/
11336  064150  043117  051506  052105  ERR108: .ASCIZ  /OFFSET =/
11337  064156  036440     000
11338  064161     110  030504  020040  ERR111: .ASCIZ  /HD1     HD2     HD3/<15><12>
11339  064166  020040  044040  031104
11340  064174  020040  020040  044040
11341  064202  031504  005015     000
11342  064207     104  052101  020101  ERR200: .ASCIZ  /DATA LATE WHEN UNLOADING HEADER/
11343  064214  040514  042524  053440
11344  064222  042510  020116  047125
11345  064230  047514  042101  047111
11346  064236  020107  042510  042101
11347  064244  051105     000
11348  064247     103  047117  051124  ERR201: .ASCIZ  /CONTROLLER ERROR DURING DRIVE SERVICING/
11349  064254  046117  042514  020122
11350  064262  051105  047522  020122
11351  064270  052504  044522  043516
11352  064276  042040  044522  042526
11353  064304  051440  051105  044526
11354  064312  044503  043516     000
11355  064317     105  051122  051117  ERR202: .ASCIZ  /ERROR DURING ERROR RECOVERY/<15><12>
11356  064324  042040  051125  047111
11357  064332  020107  051105  047522
11358  064340  020122  042522  047503
11359  064346  042526  054522  005015
11360  064354     000
```

```
11361  064355     105  051122  051117  ERR203: .ASCIZ  /ERROR RECOVERY SUCESSFUL/<15><12>
11362  064362  051040  041505  053117
11363  064370  051105  020131  052523
11364  064376  042503  051523  052506
11365  064404  006514  000012
11366  064410  051105  047522  020122  ERR204: .ASCIZ  /ERROR RECOVERY UNSUCESSFUL/<15><12>
11367  064416  042522  047503  042526
11368  064424  054522  052440  051516
11369  064432  041525  051505  043123
11370  064440  046125  005015     000
11371  064445     105  051122  051117  ERR205: .ASCIZ  /ERROR THRESHOLD EXCEEDED/<15><12>
11372  064452  052040  051110  051505
11373  064460  047510  042114  042440
11374  064466  041530  042505  042504
11375  064474  006504  000012
11376  064500  052521  051505  044524  ERR206: .ASCIZ  /QUESTIONABLE DRIVE STATUS/<15><12>
11377  064506  047117  041101  042514
11378  064514  042040  044522  042526
11379  064522  051440  040524  052524
11380  064530  006523  000012
11381  064534  047503  052116  047522  ERR207: .ASCIZ  /CONTROLLER TIMED OUT WITH GO SET/<15><12>
11382  064542  046114  051105  052040
11383  064550  046511  042105  047440
11384  064556  052125  053440  052111
11385  064564  020110  047507  051440
11386  064572  052105  005015     000
11387  064577     105  051122  051117  ERR208: .ASCIZ  /ERROR WHILE DIAGNOSING/<15><12>
11388  064604  053440  044510  042514
11389  064612  042040  040511  047107
11390  064620  051517  047111  006507
11391  064626  000012
11392  064630     127                  ERR209: .BYTE   127
11393  064631     110  040505  042504          .ASCIZ  /HEADER ERROR/<15><12>
11394  064636  020122  051105  047522
11395  064644  006522  000012
11396  064650  025052  043052  051111  ERR210: .ASCIZ  /***FIRST ERROR INFO***/<15><12>
11397  064656  052123  042440  051122
11398  064664  051117  044440  043116
11399  064672  025117  025052  005015
11400  064700     000
11401  064701     122  050113  051517  ERR211: .ASCIZ  /RKPOS /
11402  064706  000040
11403  064710  045522  040520  020124  ERR212: .ASCIZ  /RKPAT /
11404  064716     000
11405  064717     101  042104  042522  ERR213: .ASCIZ  /ADDRESS OF BEGINNING OF SECTOR IN ERROR /
11406  064724  051523  047440  020106
11407  064732  042502  044507  047116
11408  064740  047111  020107  043117
11409  064746  051440  041505  042524
11410  064754  020122  047111  042440
11411  064762  051122  051117  000040
11412  064770  005015  040504  040524  ERR214: .ASCIZ  <15><12>/DATA READ/<15><12>
11413  064776  051040  040505  006504
11414  065004  000012
11415  065006  045522  051503  020061  ERR300: .ASCIZ  /RKCS1   RKCS2   RKWCR   RKBA    RKDA    RKDC    RKASOF  RKER/<15><12>
11416  065014  020040  045522  051503
```

```
11417  065022  020062  020040  045522
11418  065030  041527  020122  020040
11419  065036  045522  040502  020040
11420  065044  020040  045522  040504
11421  065052  020040  020040  045522
11422  065060  041504  020040  020040
11423  065066  045522  051501  043117
11424  065074  020040  045522  051105
11425  065102  005015     000
11426  065105     122  042113  020123  ERR301: .ASCIZ  /RKDS    RKMR1   RKMR2   RKMR3   RKPOS   RKPAT/<15><12>
11427  065112  020040  051040  046513
11428  065120  030522  020040  051040
11429  065126  046513  031122  020040
11430  065134  051040  046513  031522
11431  065142  020040  051040  050113
11432  065150  051517  020040  051040
11433  065156  050113  052101  005015
11434  065164     000
11435  065165     122  042113  006523  ERR302: .ASCIZ  /RKDS/<15><12>
11436  065172  000012
11437  065174  005015  042515  051523  ERR303: .ASCIZ  <15><12>/MESS       A       B/<15><12>/ 00     /
11438  065202  020040  020040  020040
11439  065210  040440  020040  020040
11440  065216  020040  041040  005015
11441  065224  030040  020060  020040
11442  065232  020040     000
11443  065235     015  020012  030460  ERR304: .ASCIZ  <15><12>/ 01     /
11444  065242  020040  020040  000040
11445  065250  005015  030440  020060  ERR305: .ASCIZ  <15><12>/ 10     /
11446  065256  020040  020040     000
11447  065263     015  020012  030461  ERR306: .ASCIZ  <15><12>/ 11     /
11448  065270  020040  020040  000040
11449  065276  047125  054105  042520  ERR400: .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
11450  065304  052103  042105  046440
11451  065312  046505  051117  020131
11452  065320  040520  044522  054524
11453  065326  052040  040522  000120
11454  065334  044524  042515     072  TIM000: .ASCII  /TIME:/
11455  065341     040  020040  020072  TIM001: .ASCIZ  /   :  :  /<15><12>
11456  065346  035040  020040  005015
11457  065354     000
11458          065356                  .EVEN
```

```
11459                                   .SBTTL  *** DUMP MEMORY ***
11460
11461                                   .EVEN
11462  065356  000040                           .BLKW   40              ;RESERVE SPACE FOR STACK
11463  065456  000000                   ..STCK: .WORD   0               ;STORED STACK POINTER
11464
11465  065460  012737  000340  177776  ..DUMP: MOV     #PR7,PS         ;LOCK OUT ALL INTERRUPTS
11466  065466  010637  065456                   MOV     SP,..STCK       ;STORE STACK POINTER
11467  065472  012706  065456                   MOV     #..STCK,SP      ;LOAD STACK POINTER
11468  065476  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11469  065500  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11470  065502  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11471  065504  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11472  065506  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11473  065510  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11474  065512  013705  000224                   MOV     ..LOW,R5        ;STORE FIRST ADDRESS
11475  065516  004737  066050                   JSR     PC,..CR         ;TYPE <CR><LF>
11476  065522  004037  066130                   JSR     R0,..ASCI       ;TYPE R0 =
11477  065526  066236                           ..R0
11478  065530  011646                           MOV     (SP),-(SP)
11479  065532  004737  066154                   JSR     PC,..OCT
11480  065536  004737  066050                   JSR     PC,..CR
11481  065542  004037  066130                   JSR     R0,..ASCI       ;TYPE R1 =
11482  065546  066251                           ..R1
11483  065550  016646  000002                   MOV     2(SP),-(SP)
11484  065554  004737  066154                   JSR     PC,..OCT
11485  065560  004737  066050                   JSR     PC,..CR
11486  065564  004037  066130                   JSR     R0,..ASCI       ;TYPE R2 =
11487  065570  066264                           ..R2
11488  065572  016646  000004                   MOV     4(SP),-(SP)
11489  065576  004737  066154                   JSR     PC,..OCT
11490  065602  004737  066050                   JSR     PC,..CR
11491  065606  004037  066130                   JSR     R0,..ASCI       ;TYPE R3 =
11492  065612  066277                           ..R3
11493  065614  016646  000006                   MOV     6(SP),-(SP)
11494  065620  004737  066154                   JSR     PC,..OCT
11495  065624  004737  066050                   JSR     PC,..CR
11496  065630  004037  066130                   JSR     R0,..ASCI       ;TYPE R4 =
11497  065634  066312                           ..R4
11498  065636  016646  000010                   MOV     10(SP),-(SP)
11499  065642  004737  066154                   JSR     PC,..OCT
11500  065646  004737  066050                   JSR     PC,..CR
11501  065652  004037  066130                   JSR     R0,..ASCI       ;TYPE R5 =
11502  065656  066325                           ..R5
11503  065660  016646  000012                   MOV     12(SP),-(SP)
11504  065664  004737  066154                   JSR     PC,..OCT
11505  065670  004737  066050                   JSR     PC,..CR
11506  065674  004037  066130                   JSR     R0,..ASCI       ;TYPE SP =
11507  065700  066340                           ..SP
11508  065702  013746  065456                   MOV     ..STCK,-(SP)
11509  065706  004737  066154                   JSR     PC,..OCT
11510  065712  004737  066050                   JSR     PC,..CR
11511  065716  004737  066050                   JSR     PC,..CR
11512  065722  004037  066130                   JSR     R0,..ASCI       ;TYPE HEADER
11513  065726  066353                           ..ADD
11514  065730  004737  066050                   JSR     PC,..CR
```

```
11515  065734  004737  066050                  JSR     PC,..CR
11516
11517  065740  010546                  10$:    MOV     R5,-(SP)        ;TYPE ADDRESS
11518  065742  004737  066154                  JSR     PC,..OCT
11519  065746  012703  000004                  MOV     #4,R3           ;TYPE 4 LOCATIONS PER LINE
11520  065752  004037  066130          12$:    JSR     R0,..ASCI
11521  065756  066376                          ..BLNK
11522  065760  012546                          MOV     (R5)+,-(SP)     ;TYPE CONTENTS
11523  065762  004737  066154                  JSR     PC,..OCT
11524  065766  005303                          DEC     R3              ;CHECK IF START OF NEXT LINE
11525  065770  001370                          BNE     12$
11526  065772  004737  066050                  JSR     PC,..CR
11527  065776  005777  113136                  TST     @SWR            ;CHECK IF STOP PRINT OUT
11528  066002  100001                          BPL     15$
11529  066004  000000                          HALT
11530  066006  023705  000226          15$:    CMP     ..HIGH,R5       ;CHECK IF FINISHED
11531  066012  103352                          BHIS    10$
11532  066014  004737  066050                  JSR     PC,..CR
11533  066020  004737  066050                  JSR     PC,..CR
11534  066024  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11535  066026  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11536  066030  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11537  066032  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11538  066034  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11539  066036  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
11540  066040  013706  065456                  MOV     ..STCK,SP       ;RESTORE STACK POINTER
11541  066044  000000                          HALT
11542  066046  000776                          BR      .-2
11543
11544  066050  105777  113074          ..CR:   TSTB    @$TPS           ;WAIT FOR READY
11545  066054  100375                          BPL     ..CR
11546  066056  112777  000015  113066          MOVB    #15,@$TPB       ;LOAD <CR>
11547  066064  105777  113060          5$:     TSTB    @$TPS           ;WAIT FOR READY
11548  066070  100375                          BPL     5$
11549  066072  112777  000012  113052          MOVB    #12,@$TPB       ;LOAD <LF>
11550  066100  113701  001155                  MOVB    $FILLS,R1       ;LOAD FILL COUNT
11551  066104  001410                          BEQ     20$             ;RETURN IF NO FILLS
11552  066106  105777  113036          10$:    TSTB    @$TPS           ;WAIT FOR READY
11553  066112  100375                          BPL     10$
11554  066114  113777  001154  113030          MOVB    $NULL,@$TPB     ;LOAD FILL CHARACTER
11555  066122  005301                          DEC     R1              ;CHECK IF FILL COMPLETE
11556  066124  001370                          BNE     10$
11557  066126  000207                  20$:    RTS     PC              ;RETURN
11558
11559  066130  012001                  ..ASCI: MOV     (R0)+,R1        ;STORE ADDRESS OF ASCII
11560
11561  066132  105711                  1$:     TSTB    (R1)            ;CHECK IF FINISHED
11562  066134  001406                          BEQ     20$
11563  066136  105777  113006          2$:     TSTB    @$TPS           ;WAIT FOR READY
11564  066142  100375                          BPL     2$
11565  066144  112177  113002                  MOVB    (R1)+,@$TPB     ;TYPE CHARACTER
11566  066150  000770                          BR      1$
11567
11568  066152  000200                  20$:    RTS     R0              ;RETURN
11569
11570  066154  012701  000006          ..OCT:  MOV     #6,R1           ;LOAD DIGIT COUNT INTO R1
```

```
11571  066160  005002                      CLR     R2              ;CLEAR R2 FOR OUTPUT
11572  066162  000407                      BR      10$             ;GET LEADING DIGIT
11573
11574  066164  005002              5$:     CLR     R2              ;GET OCTAL DIGIT
11575  066166  006366  000002              ASL     2(SP)
11576  066172  006102                      ROL     R2
11577  066174  006366  000002              ASL     2(SP)
11578  066200  006102                      ROL     R2
11579  066202  006366  000002      10$:    ASL     2(SP)
11580  066206  006102                      ROL     R2
11581  066210  052702  000060              BIS     #60,R2          ;MAKE IT ASCII
11582  066214  105777  112730      15$:    TSTB    @$TPS           ;WAIT FOR READY
11583  066220  100375                      BPL     15$
11584  066222  110277  112724              MOVB    R2,@$TPB        ;TYPE OCTAL DIGIT
11585  066226  005301                      DEC     R1              ;CHECK IF WHOLE WORD PRINTED
11586  066230  001355                      BNE     5$
11587  066232  012616                      MOV     (SP)+,(SP)      ;ADJUST RETURN
11588  066234  000207                      RTS     PC              ;RETURN
11589
11590  066236  030122  036440  020040  ..R0:   .ASCIZ  /R0 =      /
11591  066244  020040  020040     000
11592  066251     122  020061  020075  ..R1:   .ASCIZ  /R1 =      /
11593  066256  020040  020040  000040
11594  066264  031122  036440  020040  ..R2:   .ASCIZ  /R2 =      /
11595  066272  020040  020040     000
11596  066277     122  020063  020075  ..R3:   .ASCIZ  /R3 =      /
11597  066304  020040  020040  000040
11598  066312  032122  036440  020040  ..R4:   .ASCIZ  /R4 =      /
11599  066320  020040  020040     000
11600  066325     122  020065  020075  ..R5:   .ASCIZ  /R5 =      /
11601  066332  020040  020040  000040
11602  066340  050123  036440  020040  ..SP:   .ASCIZ  /SP =      /
11603  066346  020040  020040     000
11604  066353     101  042104  042522  ..ADD:  .ASCIZ  /ADDRESS   CONTENTS/
11605  066360  051523  020040  041440
11606  066366  047117  042524  052116
11607  066374  000123
11608  066376  020040  020040     000  ..BLNK: .ASCIZ  /    /
11609          066404                  .EVEN
11610          066404                  HDBUFF=.
11611          066610                  BUFADD=.+132.
11612  066404  044124  020105  045522  HELP:   .ASCII  /THE RK06-RK07-K MUST BE PROPERLY FORMATED 22 SECTORS PER TRACK/<15><12>
11613  066412  033060  051055  030113
11614  066420  026467  020113  052515
11615  066426  052123  041040  020105
11616  066434  051120  050117  051105
11617  066442  054514  043040  051117
11618  066450  040515  042524  020104
11619  066456  031062  051440  041505
11620  066464  047524  051522  050040
11621  066472  051105  052040  040522
11622  066500  045503  005015
11623  066504  047524  041040  043505          .ASCII  /TO BEGIN PERFORMANCE TESTING TYPE CONTROL-C <^C>/<15><12>
11624  066512  047111  050040  051105
11625  066520  047506  046522  047101
11626  066526  042503  052040  051505
```

```
11627  066534  044524  043516  052040
11628  066542  050131  020105  047503
11629  066550  052116  047522  026514
11630  066556  020103  057074  037103
11631  066564  005015
11632  066566  042514  040507  020114          .ASCII  /LEGAL COMMANDS ARE:/<15><12><12>
11633  066574  047503  046515  047101
11634  066602  051504  040440  042522
11635  066610  006472  005012
11636  066614  047124  026440  020040          .ASCII  /TN -  TEST DRIVE N/<15><12>
11637  066622  042524  052123  042040
11638  066630  044522  042526  047040
11639  066636  005015
11640  066640  047127  026440  020040          .ASCII  /WN -  WRITE PACK,VERFY PACK, AND TEST DRIVE N/<15><12>
11641  066646  051127  052111  020105
11642  066654  040520  045503  053054
11643  066662  051105  054506  050040
11644  066670  041501  026113  040440
11645  066676  042116  052040  051505
11646  066704  020124  051104  053111
11647  066712  020105  006516     012
11648  066717     120  020116  020055          .ASCII  /PN -  CHANGE PARAMETER & TEST DRIVE N/<15><12>
11649  066724  041440  040510  043516
11650  066732  020105  040520  040522
11651  066740  042515  042524  020122
11652  066746  020046  042524  052123
11653  066754  042040  044522  042526
11654  066762  047040  005015
11655  066766  047123  026440  020040          .ASCII  /SN -  GET STATISTICS ON DRIVE N/<15><12>
11656  066774  042507  020124  052123
11657  067002  052101  051511  044524
11658  067010  051503  047440  020116
11659  067016  051104  053111  020105
11660  067024  006516     012
11661  067027     104  020116  020055          .ASCII  /DN -  DROP DRIVE N AND GET STATISTICS/<15><12><12>
11662  067034  042040  047522  020120
11663  067042  051104  053111  020105
11664  067050  020116  047101  020104
11665  067056  042507  020124  052123
11666  067064  052101  051511  044524
11667  067072  051503  005015     012
11668  067077     116  041440  047101          .ASCII  /N CAN BE 0-7 OR A/<15><12><12>
11669  067104  041040  020105  026460
11670  067112  020067  051117  040440
11671  067120  005015     012
11672  067123     104  052101  020101          .ASCII  /DATA PATTERNS MUST BE WRITTEN BY THE EXERCISER/<15><12>
11673  067130  040520  052124  051105
11674  067136  051516  046440  051525
11675  067144  020124  042502  053440
11676  067152  044522  052124  047105
11677  067160  041040  020131  044124
11678  067166  020105  054105  051105
11679  067174  044503  042523  006522
11680  067202     012
11681  067203     102  043105  051117          .ASCII  /BEFORE PERFORMANCE TESTING/<15><12><12>
11682  067210  020105  042520  043122
```

```
 11683  067216  051117  040515  041516
 11684  067224  020105  042524  052123
 11685  067232  047111  006507  005012
 11686  067240  025052  053452  051101          .ASCII  /***WARNING*** IF PROCESSOR HALT DURING DATA TRANSFER/<15><12>
 11687  067246  044516  043516  025052
 11688  067254  020052  043111  050040
 11689  067262  047522  042503  051523
 11690  067270  051117  044040  046101
 11691  067276  020124  052504  044522
 11692  067304  043516  042040  052101
 11693  067312  020101  051124  047101
 11694  067320  043123  051105  005015
 11695  067326  020040  020040  052040          .ASCII  /     TO DRIVE, BAD ECC MAY BE WRITTEN ON PACK./<15><12>
 11696  067334  020117  051104  053111
 11697  067342  026105  041040  042101
 11698  067350  042440  041503  046440
 11699  067356  054501  041040  020105
 11700  067364  051127  052111  042524
 11701  067372  020116  047117  050040
 11702  067400  041501  027113  005015
 11703  067406  020040  020040  042040          .ASCIZ  /     DRIVES MUST BE DROPPED BY PROGRAM OR WITH DROP COMMAND/<15><12><12
 11704  067414  044522  042526  020123
 11705  067422  052515  052123  041040
 11706  067430  020105  051104  050117
 11707  067436  042520  020104  054502
 11708  067444  050040  047522  051107
 11709  067452  046501  047440  020122
 11710  067460  044527  044124  042040
 11711  067466  047522  020120  047503
 11712  067474  046515  047101  006504
 11713  067502  005012     000
 11714          000001                  .END
```

```
ABASE = 177440          1135#  1704   1745
ABNORM  025470          1833   4199   4208   5936   5941   6123#  6443
ABN000  025556          6138   6142#  6193   6198   6201
ABN001  025616          6140   6144   6154#
ACDW1 = 000000          1704   1747
ACDW2 = 000000          1704   1748
ACLO  = 000010          1262#  5933   7867   7899   7982
ACPUOP= 000000          1704   1719
ADDW0 = 000000          1704   1749
ADDW1 = 000000          1704   1750
ADDW10= 000000          1704   1759
ADDW11= 000000          1704   1760
ADDW12= 000000          1704   1761
ADDW13= 000000          1704   1762
ADDW14= 000000          1704   1763
ADDW15= 000000          1704   1764
ADDW2 = 000000          1704   1751
ADDW3 = 000000          1704   1752
ADDW4 = 000000          1704   1753
ADDW5 = 000000          1704   1754
ADDW6 = 000000          1704   1755
ADDW7 = 000000          1704   1756
ADDW8 = 000000          1704   1757
ADDW9 = 000000          1704   1758
ADEVCT= 000000          1704   1710
ADEVM = 000000          1704   1746
AENV  = 000000          1704   1715
AENVM = 000000          1704   1716
AFA 9L= 000000          1704   1707
AMADR1= 000000          1704   1732
AMADR2= 000000          1704   1736
AMADR3= 000000          1704   1739
AMADR4= 000000          1704   1742
AMAMS1= 000000          1704   1726
AMAMS2= 000000          1704   1734
AMAMS3= 000000          1704   1737
AMAMS4= 000000          1704   1740
AMSGAD= 000000          1704   1712
AMSGLG= 000000          1704   1713
AMSGTY= 000000          1704   1706
AMTYP1= 000000          1704   1727
AMTYP2= 000000          1704   1735
AMTYP3= 000000          1704   1738
AMTYP4= 000000          1704   1741
APASS = 000000          1704   1709
APRIOR= 000240          1133#  1704
APTCSU= 000040          9970   10708#
APTENV= 000001          9963   10664  10706#
APTSIZ= 000200          3335   10705#
APTSPO= 000100          9965   10666  10707#
ASNORM  013662          4165#  5971   7715
ASN1$   013756          4179#  4197
ASN2$   014010          4178   4185#
ASN3$   014040          4183   4189#  4214   4237   4262   4286   4298   4387   4400   4432   4482
ASN4$   014050          4166   4193#
ASWREG= 000000          1704   1717
```

```
ATESTN= 000000  1704  1708
AUNIT = 000000  1704  1711
AUSWR = 000000  1704  1718
AVAILQ  001314  1771# 3381  3783  4350  4517  5981  6944  7697  8266  8403
AVECT1= 120210  1134# 1704  1743
AVECT2= 000000  1704  1744
A.ABNL  001456  1833# 8665  8836  8853  9047  9067  9544
A.CONT  001460  1834# 8732  8815  8824  8895  8917  8943  8988  9035  9059  9091  9140  9192
                9219  9357  9555  9601  9610  9623
A.NORM  001454  1832# 8785  8882  9003  9551  9631
BAI   = 000020  1221# 9505
BA16  = 000400  1197# 6064  7132
BA17  = 001000  1198# 6064  7132
BDSECT  042534  7320  8334#
BINOCT  053224  3531  6222  6227  6230  6233  6236  6239  6242  6245  6248  6269  6544  6549
                6637  6640  6644  6648  6651  6657  6671  6674  6678  6682  6688  6691  6694
                6700  6821  6824  6830  6833  6836  7414  7418  7592  7618  7621  7625  7634
                7661  7844  7847  9937  10050# 10233
BIT0  = 000001  1118# 1194  1211  1237  1259  1440  1841  4030  4056  4081  4156  4165  4171
                5916  6383  6484  6502  6623  6724  6773  6783  6815  6841  6911  7074  7495
                7643  7713  9608
BIT00 = 000001  1108# 1118
BIT01 = 000002  1107# 1117
BIT02 = 000004  1106# 1116
BIT03 = 000010  1105# 1115
BIT04 = 000020  1104# 1114
BIT05 = 000040  1103# 1113
BIT06 = 000100  1102# 1112
BIT07 = 000200  1101# 1111
BIT08 = 000400  1100# 1110
BIT09 = 001000  1099# 1109
BIT1  = 000002  1117# 1212  1239  1441  1842  4188  4193  4195  6061  6487  6733  6775  6781
                6803  7074  7491  7493  7713
BIT10 = 002000  1098# 1199  1228  1250  1285  1299  1312  1327  1341  1453  1851  4198  7072
                7883  7937
BIT11 = 004000  1097# 1200  1229  1251  1270  1286  1300  1313  1328  1342  1454  4206  7078
                7112  7113  7289  7883  7942  8007  8009  8071  8160
BIT12 = 010000  1096# 1202  1230  1252  1287  1301  1314  1315  1329  1343  1455  7077  7112
                8010  8016  8018
BIT13 = 020000  1095# 1203  1231  1253  1271  1288  1302  1316  1330  1344  1456  7077  7112
                8026  8032  8034
BIT14 = 040000  1094# 1204  1232  1254  1272  1289  1303  1317  1331  1345  1457  1852  8205
                8249  8334  8388
BIT15 = 100000  1093# 1205  1206  1233  1255  1273  1290  1350  1458  1853  6154  6513  6843
                6863  6881  7009  7341  7505  7860  8358  8380  8408
BIT2  = 000004  1116# 1213  1240  1241  1261  1442  1843  4182  4201  4203  6073  6152  6490
                6742  6786  6791  6812  7074  7482  7713  8234  8427
BIT3  = 000010  1115# 1219  1220  1242  1262  1443  1844  4219  4253  4267  4300  4316  5935
                6113  6493  6526  6528  6838  6923  7713  7883  7948  8337
BIT4  = 000020  1114# 1214  1221  1243  1263  1264  1279  1321  1335  1444  1845  3439  4032
                4058  4330  4393  6192  6826  6986  7309  8077  8081  8431  8435  8444
BIT5  = 000040  1113# 1222  1223  1244  1265  1280  1294  1307  1322  1336  1445  1846  4406
                4461  4475  5947  6200  6735  6805  7007  8447  9434
BIT6  = 000100  1112# 1195  1224  1245  1266  1281  1295  1308  1323  1337  1446  1848  4213
                4216  4218  6148  6739  6809  7010  7019  7713  8056  8059  8431
BIT7  = 000200  1111# 1196  1225  1246  1267  1268  1282  1296  1309  1324  1338  1448  4304
```

```
                    4307   5940   6139   6154   6843   6863   6881   6925   6955   6967   7341   7505   7860
                    8358   8380   8408  10209
BIT8  = 000400      1110#  1197   1226   1247   1248   1269   1283   1297   1310   1325   1339   1449   1849
                    6197   7001   7072
BIT9  = 001000      1109#  1198   1227   1249   1284   1298   1311   1326   1340   1450   1850   7072
BLANKS  061126      6228   6234   6240   6246   6272   6554   6638   6641   6645   6649   6654   6672   6675
                    6679   6683   6689   6692   6697   6822   6831   6834   7617   7619   7622   7631   7845
                   11027#
BPTVEC= 000014      1125#
BSBA    001662      1952#  8255   8325*  8342   8372
BSCYLN  001656      1950#  8254   8273   8291*  8300*  8301   8370
BSE   = 000200      1246#  5914   6425   7318   8912   9204
BSSECT  001654      1948#  8253   8272   8290*  8292*  8293   8295*  8315   8371
BSTRAN  042320      8250   8290#  8339
BSTRCK  001655      1949#  8296*  8297   8299*  8307
BSWC    001660      1951#  8251   8329*  8352   8368
BSWORD  001664      1953#  8322*  8359   8363   8383   8386
BUFADD= 066610      3371   8093  11611#
BUFER1  023462      5660#  5997   8227   8361
BUFFAL  022616      3801   4233   4258   4383   5376   5412#
BUFREL  023020      4277   4289   4329   4484   4491   4515   5491#  6011   6747   6889   6910   6995   7671
                    7730   7999   8264   8278   8395   8420   8474
BWAITQ  001320      1772#  3799   3810   4230   4240   4256   4381   4390   4477   4486   5374   5379   5386
B.BA16= 000001      1211#  9499
B.BA17= 000002      1212#  9499
B.CDT = 000004      1213#  4997   5000   6445   6447   6451   6455   9181   9499
B.CFMT= 000020      1214#  9181   9499   9576
CALSEC  035770      7469   7490   7535#
CALWC   037366      7199   7272   7374   7803#
CCLR  = 100000      1206#  6514   9134
CDT   = 002000      1199#  8888   9479   9516   9566   9586   9917
CERR  = 100000      1205#  6419   8780   8843   8902   9136   9187   9527   9606
CFMT  = 010000      1202#  8888   9479   9516   9566   9586   9917
CHKADD  025210      4264   4317   5949   5989   6059#  8221   8246
CHKCLK  043550      3391   8492#
CINITQ  001324      1773#  3806   4161   4190   5202   5384   5952   5986   6007   6053   6134   6187   6520
                    6961   7016   7081   7250   7292   7305   7315   7344   7450   7478   7531   7702   7727
                    7795   7863   7992   8013   8029   8053   8074   8151   8163   8202   8243   8258   8374
                    8392   8416   9102   9726   9739   9742*  9771   9786   9789*  9922
CLEAR = 000105      1165#
CLKFLG  001374      1798#  3585   8495*  8501*  8510   8548
CLKFRQ  001353      1787#  8522*  8527*  8529*
CLKINT  043642      3740   8510#
CLOCK   043734      8520   8527#
CLR   = 000040      1222#
CMDTO = 000100      1446#  6125   8659
CMPBUF  023540      5668   5701#  7404
CNTCNT  001630      1935#  3378*  7850*  7851
CNTRL1  027342      5917   6114   6475#
COE   = 001000      1249#  5914   7117   8912   9204
CONCLR= 000176      1183#  9592
CONERR  037500      7028   7034   7040   7086   7092   7119   7128   7143   7151   7158   7392   7831#  8346
                    8354
CONHUN  044210      8567   8569   8571   8579#
CONTRL  026524      1834   6181   6332#
CPUID   001364      1793#  3489*  3699*  5618
```

```
CR     = 000015  1033#  10026  10036  10747  10753  10758
CRLF   = 000200  1034#   3346   9980  10036
CTO    = 004000  1200#   5888   7026   9200
C.INIT   050074  9396#   9881
C.OPT    052116  3762    8936   9007   9103   9819#
C.RTRN   051542  9508    9534   9542   9545   9548   9552   9556   9570   9590   9595   9602   9611   9624
                 9628    9633#
C.SPEC   050624  9436    9512#
DATERR   025602  5998    6151#
DCK    = 100000  1255#   5914   6425   6817   7347   8912   9204
DCLO   = 000020  1264#
DDISP  = 177570  1040#   1687   3323
DDT    = 000400  1269#
DECBIN   054542  3568    3594   4705   4730   4753   4795   4821  10353#
DESL   = 000010  1219#
DETOFF   037204  7238    7438   7522   7765#  8139
DI     = 040000  1204#   8934   9616   9838
DISPLA   001142  1687#   3323*  3331*
DISPRE   000174  1624#   3331
DLT    = 100000  1233#   5900   6423   7323   8870   8906   9202
DLTCNT   001626  1934#   3377*  7331*  7332
DMD    = 000040  1280#
DRA    = 000001  1259#   8802   9537   9849
DRDY   = 000200  1268#   4177   4196   8061
DROP     013162  3738    3788   4031   4057   4073#  4279   4338   4493   5361   6025   6748   6890   6929
                 6996    7673   7733   8000   8422   8475
DROPDV   012322  3865    3943#
DROT   = 000040  1265#   5933   7106
DRPAR  = 000010  1242#   5926   6409   7064
DRPDRV = 002000  1453#   3786   3950   3965   4290   4402   7717   8279   9421   9899
DRVCNT   001632  1936#   3883*  3889*  3893   3945*  3951*  3955   3981*  3987*  3991   4017*  4034*  4038
DRVDSC = 000040  1445#   7892   7961   8456   8820   8995   9064   9096
DRVHRD = 000020  1444#   7892   7954   8453   8827   9038   9209
DRVMSK = 000007  1218#   6173   8748   8763   8921   9407
DRVPDT = 000004  1442#   3430   8930   9000   9012   9794   9859   9873   9876   9903
DRVPOS = 000002  1441#   3430   7012   8930   8964   9458   9730   9794   9859   9903
DRVSTT   057124  4090   10830#
DRVSZD = 010000  1455#   3430   6190   7044   8804   8955   9084   9541   9860
DRVUSE = 000001  1440#   3432   3774   3886   3901   3922   3934   3948   3963   3984   3999   4020   4046
                 4157    5938   6131   6137   9421   9899
DRY    = 000200  1267#
DR.CLR = 000005  1190#   6516   8808   8980   9029   9053   9085   9207
DR.SEL = 000001  1189#   8878   8966   9072   9328   9335   9342   9349   9518   9844
DSC    = 040000  1272#
DSWR   = 177570  1039#   1686   3322
DTBAII = 100000  1458#   9421   9503   9899
DTE    = 010000  1252#   5914   6425   7162   9204
DTYE   = 000040  1244#   5914   6379   6425   7876   7913   9204
DTYPE    020566  4527    4982#  4995
D.AWCK = 000000  1366#
D.CERT = 000012  1367#   4719   4933
D.CTRH = 077777  1370#   4811   4937
D.CTRL = 177770  1371#   4812   4938
D.MNCL = 000000  1360#
D.MNSC = 000000  1363#
D.MNTR = 000000  1361#
```

```
D.MXSC= 000025  1364#  4621  4642  4647  4929
D.MXTR= 000002  1362#  4580  4601  4606  4927
D.RATE= 000003  1365#  4674  4931
D.SKET= 000005  135    4765  4935
D.UERT= 000005  1368#  4743  4934
D.WTHI= 077777  1372#  4844  4939
D.WTLO= 177770  1373#  4845  4940
ECCCOR  026336  6279#  7396
ECCPAT  001622  1933#  6299*  6300*  4306*  6307*  6312  6313  6319  6320
ECCW  = 020000  1288#
ECH   = 000100  1245#  5914  6425  7384  8912  9204
EC3$    040502  8008  8016#
EC4$    040726  8033  8056#
EC5$    041032  8058  8077#
EC6$    041556  8079  8185#
EC7$    042064  8208  8246#
EMTVEC= 000030  1128#
ERCONT  001564  1918#  3376*  3764  5916*  5943  6061*  6073*  6113*  6129  6183  6484  6487  6490
                6493  6502  6513*
ERCVY   032160  5966  6158  6967#  8479
ERCVY1  040440  5957  8007#
ERCVY2  043236  6146  8431#
ERRPRO  001566  1925#  3373*  5958  5965*  6155  6157*  6159  6750*  6892*  6931*  6946*  6963*  6983*
                6998*  7342*  7674*  7685*  7861*  8002*  8259*  8268*  8271*  8377*  8405*  8417*  8424*
                8480  8486*
ERRPRQ  001330  1774#  5961  6162  6168  8484
ERRVEC= 000004  1121#  3320  3321*  3332*  3350*  3351*  3363*  3364*  3623*  3624*  3626*  3627*  3631*
                3632*  8492*  8493*  8499*  8505*  8506*  10412  10413  10417*  10418*  10425*  10426*
ERR000  061131  6400  6510  7334  7853  11028#
ERR001  061156  6346  11032#
ERR002  061223  6351  11039#
ERR003  061252  6356  11043#
ERR004  061305  6362  6433  11048#
ERR005  061351  6367  11055#
ERR006  061400  6340  11059#
ERR007  061426  6511  11063#
ERR008  061470  7854  11069#
ERR009  061534  7335  11075#
ERR012  061571  7584  11080#
ERR013  061707  6486  11094#
ERR014  061746  6489  11100#
ERR015  061777  6492  11105#
ERR016  062023  6496  11109#
ERR017  062053  7029  11114#
ERR020  062107  7035  11119#
ERR021  062122  7041  11121#
ERR022  062141  7022  11124#
ERR023  062206  7053  11131#
ERR024  062225  7067  11135#
ERR025  062241  7059  11138#
ERR026  062261  6978  11142#
ERR027  062321  7956  8455  1148#
ERR028  062340  7964  8458  1151#
ERR029  062400  7047  11157#
ERR030  062430  7968  8452  11162#
ERR031  062450  7930  11165#
```

ERR032	062464	7933	11167#				
ERR033	062501	7901	11170#				
ERR034	062510	7904	11172#				
ERR035	062523	7915	11174#				
ERR036	062540	7920	11177#				
ERR037	062553	7925	11179#				
ERR038	062577	7101	11183#				
ERR039	062614	7109	11186#				
ERR040	062631	7286	11190#				
ERR042	062641	8176	11193#				
ERR043	062660	8120	11197#				
ERR044	062675	7087	11201#				
ERR045	062713	7093	11204#				
ERR046	062726	7211	11206#				
ERR047	062746	7328	11210#				
ERR048	062762	7379	11212#				
ERR050	062774	7458	11215#				
ERR051	063013	7120	11219#				
ERR052	063033	8064	8068	8437	8449	11222#	
ERR053	063111	7939	11231#				
ERR054	063152	7945	11238#				
ERR056	063220	7910	11246#				
ERR057	063233	7950	11248#				
ERR058	063274	7960	8461	11254#			
ERR059	063351	6989	8446	11262#			
ERR060	063422	7411	11269#				
ERR062	063446	7499	11273#				
ERR064	063473	8157	11277#				
ERR065	063503	7301	11280#				
ERR066	063526	7393	7840	11284#			
ERR067	063546	6819	7842	11287#			
ERR070	063567	7129	8355	11290#			
ERR071	063612	7144	8347	11294#			
ERR072	063636	7152	11298#				
ERR073	063657	7159	11301#				
ERR074	063704	7004	11305#				
ERR100	063747	6662	11312#				
ERR101	064060	6630	11326#				
ERR102	064142	6633	6667	11335#			
ERR108	064150	7658	11336#				
ERR111	064161	6828	11338#				
ERR200	064207	6373	11342#				
ERR201	064247	6381	6385	6428	11348#		
ERR202	064317	6731	6798	7502	7974	11355#	
ERR203	064355	7680	11361#				
ERR204	064410	6928	6938	8421	11366#		
ERR205	064445	6888	6918	7672	11371#		
ERR206	064500	6737	6807	11376#			
ERR207	064534	6395	11381#				
ERR208	064577	8434	11387#				
ERR209	064630	8182	8441	11392#			
ERR210	064650	8467	11396#				
ERR211	064701	7412	11401#				
ERR212	064710	7416	11403#				
ERR213	064717	6539	11405#				
ERR214	064770	6545	11412#				

```
ERR300  065006          6208   6260   11415#
ERR301  065105          6211   11426#
ERR302  065165          6220   11435#
ERR303  065174          6223   11437#
ERR304  065235          6231   11443#
ERR305  065250          6237   11445#
ERR306  065263          6243   11447#
ERR400  065276          9935   11449#
EXAREA  001616          1932#  5054*  5056*  5072   5076*  5092*  5094*  5109   5113*  5115   5117   5118   5122
                        5133   5134*  5135
E.CCLR= 000001          1841#  6343   9139
E.CERR= 001000          1850#  6376   9217
E.CLAT= 000020          1845#  6359   8813   8986   9034   9058   9090
E.CMTO= 040000          1852#  6180   6393
E.CONT  001462          1835#  3375*  6180*  6337   8731*  8813*  8823*  8894*  8916*  8942*  8986*  9034*  9058*
                        9090*  9139*  9191*  9217*  9356*  9554*  9600*  9608*  9621*
E.DLT = 000400          1849#  6370   8894
E.DPAR= 002000          1851#  6404   8823   9356
E.ILLD= 000100          1848#  6365   9600
E.MDS = 100000          1853#  6338   8731   9191   9554
E.NOAT= 000002          1842#  6349   8942
E.SCLR= 000040          1846#  9621
E.UATT= 000004          1843#
E.UDAT= 000010          1844#  6354   8916
E.UNLD= 020000          1456#  3430   6195   8831   8960   9042   9052
E0$     033144          7118   7123#
E1$     033306          7150   7157   7162#
E10$    035466          7455   7482#  8235   8428
E2$     034010          7164   7253#
E3$     034346          7255   7318#
E4$     034504          7324   7347#
E5$     035314          7349   7453#
FBLKC   001670          1958#  3365*  5419   5456*  5504   5529*  5554*
FBLKT   001672          1960#  3366   3371*  3400*  3405*  3408*  3491*  3504*  3507*  3520   3521   3700*  3704*
                        3707*  3708   3709   5421   5503
FLAG    001361          1791#  3294*  3297*  3300*  3303*  3397   3465   3727   3749   10457*
FLPARM  027206          6388   6445#
FMTE  = 000020          1243#  5914   6425   7876   7918   9204
GENERT  021336          3795   5188#
GETBUF  022520          4280   4339   4487   4518   5372#  6055   6749   6891   6930   6945   6962   6997   7675
                        7734   8001   8267   8404   8423   8476
GETCUR  037124          7013   7257   7738#
GETNUM  027252          4226   4445   4538   5333   6455#
GNS   = ****** U        1623   3345   10740  10743  10744
GO    = 000001          1194#  6178
HCRC  = 000400          1247#
HDBUF= 066404           7312   8106   11610#
HDR.AD  001502          1876#  8864   8868*  9575*
HDR.CT  001504          1877#  8872*  9578*  9581*
HEAD1   001646          1945#  6829   9107*  8112   8115   8154
HEAD2   001650          1946#  6832   8108*  8113   8114   8173
HEAD3   001652          1947#  6835   8109*  8117
HELP    066404          3744   11612#
HLTOOO  031426          6745   6845   6850#  6981   6994   7330   7507   7849   7979   8473
HOLD    027336          4446   6459*  6464*  6470#
HOLD1   027340          5334   6460*  6465*  6471#
```

```
HOUR    001366  1795# 3449* 8544* 8553
HT    = 000011  1031# 9978  10036
HVRC  = 000400  1248# 5914  6425  7253  8912  9204
HZ      001352  1785# 8522  8529
H.HEAD  001507  1880# 7297
IDAE  = 002000  1250# 5926  7876  7928
IE    = 000100  1195# 5368  9144  9212  9630  10202 10218
ILC   = 000001  1237# 5914  6425  7038  9204
ILF   = 000004  1240#
ILLCMD  056440  3874* 4994  10772#
ILLCOM  056407  3875  10767#
INTMSK  001514  1886# 4149  6431  6436  6941  6953  7234  7436  7520  7688  8137  8651  8658
                8778  8783  8811  8832  8847  8857  8860  8876  8924  8947  8984  8998  9014
                9032  9043  9056  9065  9088  9099  9408  9529  9539  9549  9853  9871  9886
INTR  = 000300  1185# 4413  5979  5993  9841
IOTVEC= 000020  1126#
IR    = 000100  1224#
I.AERR  046310  8972  8978  8996  9012# 9077  9082  9097
I.ATTN  045576  8742  8760  8900#
I.CCLR  047070  8797  8851  9134# 9210  9533  9594  9857
I.CSTS  047542  5887  8790  9285#
I.CST1  047564  8794  8850  9289# 9524
I.DUAL  046664  8957  9072#
I.ERR   044772  8791  8795#
I.ERRA  044752  8790# 8862
I.ERRC  044760  8753  8781  8793# 8932
I.HDAL  045236  8774  8841#
I.INTR  044444  3410  3725  8721#
I.ISRL  001506  1879# 3396* 4412* 5978* 5992* 8735  8738* 8741* 9145* 9431* 9564* 9629* 9840*
                9891*
I.ISSU  047152  6517  8809  8879  8967  8981  9030  9054  9073  9086  9176# 9329  9336  9343
                9350  9845
I.IOO   044610  8739  8757#
I.RTRN  047052  8733  8786  8798  8807  8810  8816  8825  8834  8837  8852  8854  8880  8883
                8892  8896  8918  8937  8944  8968  8982  8989  9005  9008  9028  9031  9036
                9045  9048  9055  9060  9068  9074  9087  9092  9105#
I.STAT  047646  8806  9027  9326# 9547
I.STOR  047414  6333  6418  9186  9240#
I.UNLD  046560  8962  9052#
KIPAR0= ****** U  10404
LDDATA  023234  3804  4236  4261  4386  4481  5382  5596#
LDEXAR  020664  4895  4899  4903  4907  4911  5039#
LF    = 000012  1032# 10030 10036 10747 10753 10758
LSTCYL  027332  4227  4228  4539  4560  4565  4925  4998* 5001* 5052  5090  6458* 6463* 6468#
MAIN    011420  3751  3752  3759# 3765  3785  3789  3797  3807  3812
MAXBUF  001354  1788# 3545* 3556* 3557  3559* 3570  3713* 3717* 4370  5228
MCLK  = 000400  1283#
MDS   = 001000  1227# 5900  6423  8729  9189  9531
MEMBAS= 172100  1137# 3352
MEMERR  052650  3356  9935#
MEMVEC= 000114  1136# 3356* 3357*
MERD  = 001000  1284#
MESMSK= 000017  1277#
MEWD  = 002000  1285#
MIND  = 000200  1282#
MINUTE  001370  1796# 3448* 8540* 8541  8543* 8554
```

```
MSP    = 000100          1281#
M.CADD= 017760           1353#  7278
M.CDIF= 017760           1352#
M.DRV  = 000007          1349#
M.HEAD= 007000           1356#
M.ID   = 000003          1351#
M.PAR  = 100000          1350#
M.SECT= 000760           1355#  8084
M.SER  = 077770          1354#  4173
NED    = 010000          1230#  5900   6423   7050   8744   8799   9198   9202   9535   9847
NEM    = 004000          1229#  5900   6423   7084   8906   9202
NERSTT  031456           6861#  7068   7110   8069
NEWFLG  027334           4167   4169*  6389*  6440   6469#
NOCHK  = 000400          1449#  9421   9627   9899
NODSC  = 004000          1454#  7892   7957   8459   8977   9081
NORMAL  024400           1832   5887#  6442
NORM1   024774           5996   6001#  7705
NORM2   025032           5999   6002   6004   6010#
NXF    = 000004          1241#  5926   7876   7923
OCTBIN  054406           3476   3539   3618   3651   3674   4535   4548   4577   4589   4618   4630   4659   5050
                         5068   5088   5105   10298#
OCTSTG  053320          10056  10074  10077#
OFFM12= 000260           1384#  7792
OFFM4  = 000220          1382#  7772
OFFM8  = 000240          1383#  7782
OFFP12= 000060           1379#  7787
OFFP4  = 000020          1381#  7767
OFFP8  = 000040          1380#  7777
OFFSET= 000115           1169#  7793   8027   9466
OFST   = 000004          1261#
OPI    = 020000          1253#  5914   6425   7253   8912   9204
OPISTT  031524           6878#  7287   7302   8158   8177
OPRCMD  011656           3841#  10215
OPR000  056446           3853   3908   3970   4006   4064   10773#
OPR001  056463           3735*  3845*  3847   3849   3851   3858*  3881   3898   3907*  3917   3930   3943   3960
                         3969*  3979   3996   4005*  4015   4043   4063*  10776#
OPR002  056466           3854   10777#
OPR003  056501           4345   4500   10779#
OPR004  056514           3909   4065   10781#
OPR005  056540           3971   4007   10785#
OPR006  056560           4080   10788#
OPR007  056626           3842   10795#
OPR008  056656           4492   7731   10799#
OPR010  056736           4079   4093   4344   4499   6622   6976   10807#
OPR011  056755           4077*  4078*  4091*  4092*  4342*  4343*  4497*  4498*  6620*  6621*  6974*  6975*  10810#
OPR012  056760           4337   10811#
OPR013  057005           3895   4040   10815#
OPR014  057047           3957   3993   10821#
OPR015  057071           4278   10825#
OPTFLG  001500           1875#  4155*  9196   9826*  9924*
OR     = 000200          1225#
OVLYLD  001365           1794#  3401   3501*  3517*  3701*
O.OVER  001513           1882#  3380*  4149*  6941*  6953*  7234*  7436*  7520*  7688*  8137*  9871
O.RTRN  052622           9842   9846   9858   9861   9884   9924#
O.WAIT  001464           1855#  3394*  6411*  6852   8660   8664*  8757   8776*  8846*  8859*  8874*  9459*  9502*
                         9560*  9574*  9828   9856*  9890*
```

```
PACK  = 000103          1164#   4180    4181
PARCYL  001602          1929#   4537*   4539    4550*   4551    4560*   4565*   4566*   4567    4568
PARM    016346          3921    3933    4522#
PARMDV  012212          3863    3917#
PARM0   001736          1897    1972#
PARM1   002214          1898    2065#
PARM10  004516          2717#   7741    7744    7755    7758    8466
PARM2   002472          1899    2158#
PARM3   002750          1900    2251#
PARM4   003226          1901    2344#
PARM5   003504          1902    2437#
PARM6   003762          1903    2530#
PARM7   004240          1904    2623#
PARSEC  001612          1931#   4620*   4621    4632*   4633    4642*   4647*   4648*   4649    4650
PARSRT  006642          1630    3294#
PARTRK  001606          1930#   4579*   4580    4591*   4592    4601*   4606*   4607*   4608    4609
PAR.EN= 000001          1138#   3355
PAR000  060105          4526    10926#
PAR001  060131          4524*   4525*   10930#
PAR002  060136          4528    10931#
PAR003  060161          4570    10935#
PAR004  060201          4611    10938#
PAR005  060222          4652    10941#
PAR006  060245          4676    10945#
PAR007  060277          4697    10950#
PAR008  060342          4722    10956#
PAR009  060407          4746    10963#
PAR010  060436          4767    10967#
PAR011  060503          4788    10974#
PAR012  060543          4814    10980#
PAR013  060610          4896    10987#
PAR014  060616          4900    10988#
PAR015  060625          4904    10990#
PAR016  060633          4908    10992#
PAR017  060642          4912    10994#
PAR018  060650          4870    5043    10995#
PAR019  060676          4869    10999#
PAR020  060705          5360    11001#
PAR021  061041          4849    11017#
PAR022  061064          4982    11021#
PASCNT  001356          1789#
PAT   = 000020          1279#
PATTAB  004602          1633    2748#   5620    5728    5825    7612
PAT0    004642          2748    2765#
PAT1    004742          2749    2798#
PAT10   005642          2756    3029#
PAT11   005742          2757    3062#
PAT12   006042          2758    3095#
PAT13   006142          2759    3128#
PAT14   006242          2760    3161#
PAT15   006342          2761    3194#
PAT16   006442          2762    3227#
PAT17   006542          2763    3260#
PAT2    005042          2750    2831#
PAT3    005142          2751    2864#
PAT4    005242          2752    2897#
```

```
PAT5     005342            2753    2930#
PAT6     005442            2754    2963#
PAT7     005542            2755    2996#
PBLKT    001524            1897#   3412    3450    3773    3885    3900    3920    3932    3947    3962    3983    3998    4019
                           4045    8656    8751    8926    8954
PBSVAL=  001000            1450#   9361
PCA    = 004000            1286#
PCD    = 010000            1287#
PERINV   001360            1790#   3599*   3609*   3719*   8523    8534    8538
PERIOD   001375            1799#   8523*   8536*   8538*
PGE    = 002000            1228#   5900    6423    7032    9202
PIP    = 020000            1271#
PIRQ   = 177772            1038#
PIRQVE=  000240            1132#
POSCMD   001666            1954#   7011*   8060
PRDSTT   026076            6219#   6738    6808    6980    7976    8463    8469
PRERWD   036202            7420    7503    7580#
PREXAR   020152            4848    4858    4869#   4885
PRGSRT   006676            3295    3298    3301    3305#
PRIBUF   027620            6524#   6840    7506
PRIREG   026030            6204#   6399
PRISER   027774            4094    4346    4501    6564#   6625
PRISTT   026252            6254#   6500    6732    6802    6979    7329    7838    7975    8462    8468
PRITIM   044046            4095    4347    4502    6398    6499    6626    6977    8548#   10456
PRI000   030156            6615#   6721    6792    6988    7325    7496    7832    7898    7907    7936    7953    8433
PRI001   030244            6505    6630#   6744    6814    7978
PRI002   030422            6501    6662#   6741    6811    6992    7504    7839    7977    8464    8470
PRI100   030642            6717#   7003    7021    7046    7052    7058    8063
PRI200   031032            6771#   7066    7100    7108    7210    7285    7300    7378    7457    8067    8119    8156    8173
                           8181
PRSTAT   026310            6209    6212    6261    6268#   6273
PR0    = 000000            1055#   3347
PR1    = 000040            1056#
PR2    = 000100            1057#
PR3    = 000140            1058#
PR4    = 000200            1059#
PR5    = 000240            1060#   1133    1831    3696    3722
PR6    = 000300            1061#   8521
PR7    = 000340            1062#   3305    3351    3357    3624    8493    8552    10452   10453   11465
PS     = 177776            1035#   1036    3305*   3747*   3798*   3803*   3811*   3906*   3910*   3968*   3972*   4004*   4008*
                           4062*   4066*   4073    4074*   4088    4089*   4131*   4235*   4260*   4357*   4380*   4385*   4414*
                           4474*   4480*   5359*   5362*   5372    5373*   5381*   5385*   5389*   5994*   6010*   6012*   6151*
                           8224*   8229*   8233*   8551    8552*   8556*   8642    8646*   8671*   9402    9403*   9633*   9656
                           9657*   9671*   9695    9696*   9705*   9825    9827*   9925*   10214*  10216*  11465*
PSW    = 177776            1036#
PWRSRT   007200            3348#   10459
PWRVEC=  000024            1127#   3316*   3317*   10440*  10451*  10452*
P.ASOF=  000032            1471#   8983*   8984    9024*   9295*   9468*   9469
P.ASSN=  000210            1578#   3434    3439*   3455*   4030*   4032*   4056*   4058*   4081    4083*   4085*   4136    4152*
                           4156*   4165    4171*   4182*   4188*   4193    4195*   4201    4203*   4213*   4216    4218*   4219*
                           4253    4267    4300    4304    4307*   4316*   4330    4393    4406    4461*   4475    4490*   4496*
                           4509    5969    6383    6623    6724    6921    6923    6925    6955*   7074    7397    7690    7713
                           8209    8262    8337
P.AWCK=  000077            1497#   4684*   4695*   4932*   5252
P.A00  = 000040            1474#   6225    8974*   8975    8991*   8992    9352*   9525*
P.A01  = 000044            1476#   9331*
```

```
P.A10 = 000050   1478#  9338*
P.A11 = 000054   1480#  4172  9345*
P.BAHI= 000007   1433#
P.BALO= 000010   1434#  4270  4424*  5443*  5451*  5665  6069  6690  7134  7140  7169  7243*  7261
                 7312*  7354  7443*  7473*  7527*  7538  7541  7724*  8144*  8255*  8324  8372*  9485
                 9575
P.BAR = 000024   1468#  6070  7134  7141  7467  8344  9021*  9291*
P.BFCP= 000222   1609#  5666*  5667*  5707  7401*
P.BUFF= 000145   1529#  3418*  5446*  5455*  5495  5497*
P.B00 = 000042   1475#  8969*  8970  9353*  9526*
P.B01 = 000046   1477#  9332*
P.B10 = 000052   1479#  7277  9339*
P.B11 = 000056   1481#  7297  8083  8881*  9346*
P.CERT= 000100   1498#  4707*  4719*  4933*  7667
P.CMLG= 000220   1608#  5663*  5708  7402*
P.CMND= 000001   1427#  4158*  4180*  4186*  4211*  4223*  4248*  4297*  4362*  4399*  4410  4426  4428*
                 4431*  5198*  5207*  5209  5279*  5604  5945  5947  5973  5976  5984*  5995  6001
                 6051*  6670  6959*  7011  7014*  7079*  7114*  7240*  7246  7290*  7313*  7440*  7446
                 7470*  7524*  7692  7694  7700*  7720*  7793*  8011*  8027*  8048*  8049  8060*  8072*
                 8141*  8147  8161*  8200*  8214  8216  8225  8237  8286*  8340  8381  8772  9434
                 9438  9450  9460  9466  9472  9475  9481  9489  9496  9512  9558  9572  9592
                 9598  9864  9866  9868
P.CS1 = 000016   1465#  5888  6064  6258  7026  7062  7070  7132  8793*  8849*  8887*  8888*  8890*
                 8891  9016*  9018*  9285*  9288*  9424  9478*  9479*  9481*  9493*  9496*  9498*  9499*
                 9507  9515*  9516*  9518*  9519  9521*  9522  9527  9565*  9566*  9568*  9569  9585*
                 9586*  9588*  9589  9605*  9606  9615*  9616  9906  9916*  9917*  9919*  9920
P.CS1H= 000007   1432#  4997*  5000*  6445  6447*  6451*  6455  8887  9180  9478  9498  9515  9565
                 9576  9585  9916
P.CS2 = 000020   1466#  5900  7032  7050  7056  7084  7090  7323  7453  8799  9019*  9289*  9432*
                 9457  9505*  9506  9514  9531  9535  9562*  9563  9584
P.CYLN= 000002   1428#  4227*  4249*  4295*  4365*  4422*  4433  4435  5199*  5208*  5349*  6109  6673
                 7147  7176  7181  7242*  7269  7311*  7353  7442*  7472*  7526*  7551  7723*  7804
                 8142*  8254*  8284*  8302  8370*  8885  9462  9487  9582  9914
P.DCYL= 000030   1470#  4422  4423  6110  7147  7176  7185  7260  7269  7360  8154  8273*  8291
                 9023*  9297*
P.DPAT= 000121   1506#  4252*  4369*  4473*  5206*  5251*  5611  6699
P.DRVN= 000000   1426#  4022  4048  4077  4091  4148  4342  4497  4524  6174  6412  6620  6940
                 6952  6974  7233  7435  7519  7687  8136  8769  9178  9406  9870
P.DS  = 000036   1473#  4177  4196  4204  4334  4355  5933  6221  6259  7106  7867  7899  7982
                 8061  8802  9026*  9293*  9537
P.DSTT= 000150   1549#  3420*  5189*  5955  6142  6154*  6526  6528*  6729  6775  6781*  6783*  6786
                 6791*  6796  6843*  6861  6863*  6871  6878  6881*  6911  6957*  7009*  7010*  7072
                 7077*  7078*  7112*  7113*  7289*  7309*  7339  7341*  7491  7493*  7495*  7500  7505*
                 7643  7683*  7858  7860*  7972  8007  8009*  8010*  8016  8018*  8026*  8032  8034*
                 8056  8059*  8071*  8077  8081*  8160*  8205  8249*  8334  8358*  8380*  8388*  8408*
                 8431  8435
P.DTS = 000026   1469#  4420  4421  6094  6107  7155  7174  7184  7259  7266  7359  8272*  8290
                 9022*  9292*
P.ECMP= 000117   1504#  4154*  5778*  7403*
P.EPAT= 000062   1483#  6299  6820  7390  7417  7744  7758  7843  8795*  9426  9908
P.EPOS= 000060   1482#  6288  6823  7386  7388  7413  7846  8796*
P.ER  = 000034   1472#  5914  5926  6409*  6817  7038  7064  7098  7117  7162  7253  7318  7347
                 7384  7876  7908  7913  7918  7923  7928  7984  9025*  9294*
P.ERAD= 000217   1607#  5770*  7486  7587  7593  7603  7624  7626
P.ERHD= 000214   1606#  5665*  5706  5712*  5816  5821  7399*  7488  7585  7598  7632
P.ERR = 000212   1594#  3419*  3454*  4076*  4198*  4206*  5190*  5935*  5940*  6139*  6148*  6152*  6192*
```

```
                    6197*  6200*  6958*  6967   6982*  6986   7001   7007   7019   7482   7684*  7883   7937
                    7942   7948   8234*  8427*  8444   8447   9863*
P.EXAR= 000232      1616#  3457   4143   4889   5283
P.HARD= 000200      1572#  4119   6913*  6914
P.IERC= 000120      1505#  4776*  4786*  4936*  6896   6934   7212   7380   7459   7508   7986   8121
P.LCMD= 000135      1523#  4179*  4185*  4210*  4222*  4308*  4358*  4415*  5221*  6631   6636   6652   8411*
P.LCYL= 000136      1524#  4295   4309*  4359*  4416*  5199   5222*  6639   8284   8412*
P.LDPT= 000144      1528#  4418*  5225*  6656
P.LSEC= 000140      1525#  4296   4310*  4360*  4417*  5200   5223*  6647   8285   8413*
P.LTRK= 000141      1526#  6643
P.LWC = 000142      1527#  4311*  4361*  4419*  5224*  6650   8414*
P.MNCL= 000066      1490#  4568*  4926*  5287   5294
P.MNSC= 000074      1494#  4650*  4930*  5311
P.MNTR= 000072      1492#  4609*  4928*  5297
P.MXCD= 000106      1501#  4797*  4800*  4801*  4811*  4812*  4937*  4938*  6016   6021
P.MXCL= 000070      1491#  4567*  4925*  5286   6448*  6452*
P.MXSC= 000075      1495#  4649*  4929*  5310
P.MXTR= 000073      1493#  4608*  4927*  5296
P.MXWT= 000112      1502#  4823*  4826*  4827*  4828*  4829*  4830*  4831*  4832*  4833*  4834*  4844*  4845*
                    4939*  4940*  6035   6040
P.NER = 000206      1575#  4128   6746*  6864*  6865   6993*  7969*  8472*
P.NODR= 000152      1566#  4098   4374*  4466*  4507   5242*  5243*  5265*  5266*  6016   6021
P.NOPI= 000204      1574#  4125   6880*  6882
P.NRD = 000164      1568#  4106   4377*  4378*  4469*  4470*  5270*  5271*  5272*  5276*  5277*  5278*  6029
                    6030
P.NSKI= 000202      1573#  4122   6873*  6883
P.NWRT= 000156      1567#  4102   4375*  4376*  4467*  4468*  5258*  5259*  5260*  5267*  5268*  5269*  6031
                    6033
P.OFST= 000006      1431#  6687   7660   7767*  7772*  7777*  7782*  7787*  7792*  9468
P.PKSR= 000226      1615#  4271*  4272*  4321*  4323*  6587   6589
P.PRST= 000014      1436#  3430*  3432   3453*  3774   3786   3886   3901   3922   3934   3948   3950*  3963
                    3965*  3984   3999   4020   4046   4075*  4157*  4255*  4283   4285*  4290   4364*  4396
                    4398*  4402   4409*  4430*  5211*  5264*  5938   6003   6005*  6125   6131   6137   6190
                    6195   7012*  7044   7248*  7448*  7476*  7717   7721*  7892   7954   7957   7961   7965
                    8051*  8149*  8239   8241*  8256*  8279   8367*  8390*  8450   8453   8456   8459   8659*
                    8788*  8804*  8820*  8827   8831*  8930*  8955   8960   8964*  8977*  8995*  9000   9012*
                    9038   9042*  9052*  9064*  9081*  9084*  9096*  9209*  9361*  9421*  9458*  9482*  9491
                    9497*  9503   9541*  9627   9730   9763   9794*  9859*  9860*  9873   9876*  9899*  9903*
P.QLNK= 000064      1487#  9659*  9664*  9701   9733   9736   9741   9780   9783   9788
P.RATE= 000076      1496#  4332   4671*  4674*  4931*  5246
P.RBAH= 000134      1518#
P.RBAL= 000132      1517#  4320   4424   5442*  5450*  5501   5601   7724   8326
P.RCMD= 000123      1512#  4159*  4179   4181*  4185   4187*  4210   4212*  4222   4224*  4248   4308   4358
                    4363*  4415   5207   5221   5257*  5263*  5275*  5279   6524   6665   6695   7218   7240
                    7382   7424   7440   7470   7524   7720   8048   8126   8141   8200   8411
P.RCYL= 000124      1513#  4228*  4249   4309   4359   4366*  4416   4423*  5208   5222   5295*  5325   5349
                    7723   8412
P.RDPT= 000122      1511#  4140   4252   4418   4471*  4472*  4473   5206   5225   5250*
P.RECT= 000146      1532#  3421*  5188*  6898*  6899   6956*  7682*  7988*  7989   8207   8375   8396
P.RERD= 000147      1533#  6685   7166   7214*  7215   7220   7225   7421*  7422   7426   7431   7461*  7462
                    7510*  7511   7516   7645   7650   7765   7770   7775   7780   7785   7790   8024   8035
                    8037   8039   8041   8043   8045   8047*  8123*  8124   8128   8133   8185   8189   8191
                    8193   8195   8197   8199*
P.RSEC= 000126      1514#  4229*  4250   4310   4312*  4313   4360   4368*  4417   4421*  5212   5223   5323*
                    5350   6949   7722   8409   8413
P.RTRK= 000127      1515#  5309*
```

```
P.RWC = 000130        1516# 4251  4311  4361  4419  4449  4479* 5213  5224  5416* 5498  5602  7725
                      8328  8414
P.SECC= 000176        1571# 4110  7647* 7664
P.SECT= 000004        1429# 4221* 4229  4250* 4296* 4313* 4367* 4420* 4443  5200* 5212* 5350* 6084
                      6681  7155  7174  7180  7241* 7266  7310* 7352  7441* 7471* 7525* 7556  7722*
                      7821  8090  8143* 8253* 8285* 8318  8371  8886  9463  9488  9583
P.SEEK= 000211        1590# 3794* 3796  4245  4247* 4294* 5195  5197* 5205* 7719* 8019  8283*
P.SERL= 000224        1614# 4172* 4173* 4174* 4175* 4176* 6568
P.SKET= 000104        1500# 4755* 4765* 4935* 6884
P.SMPL= 000116        1503# 4857* 4867* 4941* 5735
P.SOFF= 000174        1570# 4116  7655* 7666
P.SRRD= 000172        1569# 4113  7652* 7665
P.TRCK= 000005        1430# 4437  6097  6677  7550  7812  8310
P.UERT= 000102        1499# 4732* 4743* 4934* 6914
P.WC  = 000012        1435# 4225* 4251* 4370* 4371* 4373* 4375  4377  4379* 4450* 4460* 4464  4479
                      5213* 5241* 5415  5666  6066  6076  6693  7125  7137  7170  7244* 7245* 7262
                      7355  7444* 7445* 7474* 7475* 7528* 7529* 7542  7725* 8145* 8146* 8251* 8363*
                      8368* 8387* 9486
P.WCR = 000022        1467# 6059  7123  7125  8350  9020* 9290*
P0$   016352          4524# 4533  4575  4616  4657  4681  4703  4728  4751  4773  4793  4819  4854
                      4875  4918
P1$   020356          4532  4925#
P10$  020454          4792  4937#
P11$  020470          4818  4939#
P12$  020504          4853  4941#
P2$   020370          4574  4927#
P3$   020402          4615  4929#
P4$   020414          4656  4931#
P5$   020422          4680  4932#
P6$   020426          4702  4933#
P7$   020434          4727  4934#
P8$   020442          4750  4935#
P9$   020450          4772  4936#
QNEWSW= 000000        1153  1379
QUESMK  061113        7595  7628  11025#
Q.INIT= 040000        1457# 3430  9763  9794  9859  9876  9903
Q.POP   051644        3782  5378  5960  6161  8483  9692#
Q.PUSH  051564        3805  3809  4160  4189  4239  4349  4389  4485  4516  5201  5383  5951  5980
                      5985  6006  6052  6133  6167  6186  6519  6943  6960  7015  7080  7249  7291
                      7304  7314  7343  7449  7477  7530  7696  7701  7726  7794  7862  7991  8012
                      8028  8052  8073  8150  8162  8201  8242  8257  8265  8373  8391  8402  8415
                      9101  9654# 9921
Q.RMOV  052004        6123  6475  9763#
Q.SRCH  051720        9726# 9835
RDALHD= 000164        1182# 7313  8772  9572  9864
RDCHR = 104402        10131 10743#
RDDATA= 000121        1171# 4223  4224  5275  5604  5945  5995  6524  6631  6652  6665  6695  7218
                      7382  7424  8126  8225  8340  9866
RDGATE= 100000        1290#
RDHEAD= 000125        1173# 8890  9588
RDLIN = 104403        3843  4963  10744#
RDSTAT= 000141        1181# 4158  4159  7014  9512
RDY   = 000200        1196# 9522  9833
RECAL = 000113        1168# 4211  4212  6959  7079  7114  7290  8072  8161  9475
RELEAS= 000140        1180# 4410  4431  5976  5984  6051  7694  7700  8216  9558
RELFLG  001406        1804#
```

```
RESTRT  006652  1628  3297#
RESTO   007644  3398  3445#
REST1   011244  3686  3697  3724#
REST2   011260  3443  3727#
RESVEC= 000010  1122#
RKASOF= 000016  1149#  8782  8856  8939  9247  9295  9469*
RKBA  = 000004  1144#  9243  9291  9485*
RKBAS   001446  1829#  3628* 3642* 3720* 8647  8727  9413  9830
RKCS1 = 000000  1142#  4413* 5979* 5993* 6177  6416  6514* 6855  8752  8779  8841  8891* 8900
                9134* 9135  9144* 9183* 9184  9212* 9240  9507* 9519* 9521  9569* 9589* 9605
                9615  9630* 9832  9841* 9920*
RKCS2 = 000010  1146#  6172  6414* 6415* 8728  8869  9179* 9241  9289  9457* 9506* 9514* 9563*
                9584* 9604* 9915*
RKDA  = 000006  1145#  8886* 9244  9292  9463* 9488* 9583*
RKDB  = 000024  1152#  8865  8866  8867
RKDC  = 000020  1150#
RKDCYL= 000020  1151#  8885* 9248  9297  9462* 9487* 9582* 9914*
RKDS  = 000012  1147#  9245  9293
RKECPS= 000030  1157#  8796  9252
RKECPT= 000032  1159#  8795  9253
RKER  = 000014  1148#  8909  9246  9294
RKMR1 = 000026  1153#  6413* 6515* 8877* 8965* 9249  9326* 9333* 9340* 9347* 9433* 9913*
RKMR2 = 000034  1154#  9250  9525
RKMR3 = 000036  1155#  9251  9526
RKPAT = 000032  1158#
RKPOS = 000030  1156#
RKPRI   001452  1831#  3411  3685* 3696* 3722* 3726  3746  3798  3906  3968  4004  4062  4074
                4089  4380  4474  5359  5373  5385  6010  6151  8229  8233  8646  9403  9657
                9696  9827  10200 10216
RKVEC   001450  1830#  3409  3655* 3666* 3721* 3724
RKWC  = 000002  1143#  9242  9290  9486*
RLS   = 000010  1220#  9562
RSTAT   037154  7752#  8111
RTYBA   001634  1938#  6283  6541  7169* 7243  7261* 7354* 7399  7443  7467* 7468* 7473  7489*
                7527  7537  7540* 7541* 7827* 8144
RTYCMD  001645  1944#
RTYCYL  001642  1942#  7181* 7185* 7198* 7242  7260* 7283  7311  7353* 7360* 7373* 7442  7472
                7526  7551* 7570* 7803  8142
RTYSCN  001644  1943#  3372* 6909* 7179* 7200* 7208* 7227  7258* 7351* 7574* 7663* 7681*
RTYSEC  001640  1940#  7180* 7184* 7186  7188* 7191* 7229  7241  7259* 7310  7352* 7359*  7361
                7363* 7366* 7441  7471  7525  7564* 7818  8143  8171
RTYTRK  001641  1941#  7192  7194* 7197* 7296  7367  7369* 7372* 7550* 7561* 7566  7573*  7809
                8167
RTYWC   001636  1939#  6284  6542  7170* 7171* 7172  7178* 7201  7203* 7206  7244  7262*  7263*
                7264  7273  7276* 7355* 7356* 7357  7375  7377* 7401  7402  7444  7474  7528
                7549* 7825* 8145
SAVSWR  001404  1803#  10439* 10458
SCLR  = 000040  1223#  6415  9604
SECOND  001372  1797#  3447* 8530* 8531  8533* 8555
SEEK  = 000117  1170#  4297  5198  5973  7692  8011  8214  8286  9450  9460  9919
SELDRV= 000101  1163#  9568
SKI   = 000002  1239#  5926  7098
SKISTT  031510  6871#  7102
SOFCMP  001362  1792#  3572* 3583* 3718* 4847  5663
SPAR  = 020000  1203#  5888  7062  7070  9200
SPDLSS= 000020  1263#  5933  7867  7982
```

```
SRTSPL= 000111  1167#  4186  4187  9472
STACK = 001100  1026#  3312  3759  10454
START   006670  1627   3303#
START$  007274  3365#
START1  006662  1635   3300#
STAT    013260  3776   3986  4001  4088#
STATDV  012502  3868   3979#
STATIS  001633  1937#  3379* 3770  3780*  8539*
STAT00  013272  4086   4090#
STAT01  013530  4084   4131#
STKLMT= 177774  1037#
STT000  057153  6586  10834#
STT001  057202  4096  10840#
STT002  057224  4100  10843#
STT003  057251  4104  10847#
STT004  057273  4108  10851#
STT005  057326  4112  10856#
STT006  057342  4115  10858#
STT007  057356  4118  10860#
STT008  057402  4121  10864#
STT009  057426  4124  10868#
STT010  057474  4127  10875#
STT011  057514  6608  10878#
STT012  057541  6591  10882#
STT013  057554  6606  10883#
STT1$   031536  6874   6881#
STT100  057176  6567  10838#
STT2$   031562  6866   6886#
STT5$   031622  6862   6867  6872  6879  6885  6887  6896#
SUBCLR= 000177  1184#  9598
SURECY  036454  7406   7643#  8021  8211  8218  8230  8399
SVAL  = 100000  1273#
SWR     001140  1686#  3310  3322* 3324  3330* 3337* 3767  3791  4243  5192  5660  6013  6334
                6477   6480  6482  6497  6506  6508  6529  6531  6615  6618  6727  6793  6850
                6886   6916  6936  6969  6972  6990  7326  7409  7497  7656  7669  7678  7833
                7970   7980  8274  8442  10226 10255* 10439 10458* 11527
SWREG   000176  1625#  3330  10226  10232
SW0   = 000001  1090#
SW00  = 000001  1080#  1090
SW01  = 000002  1079#  1089
SW02  = 000004  1078#  1088
SW03  = 000010  1077#  1087
SW04  = 000020  1076#  1086
SW05  = 000040  1075#  1085
SW06  = 000100  1074#  1084
SW07  = 000200  1073#  1083
SW08  = 000400  1072#  1082
SW09  = 001000  1071#  1081
SW1   = 000002  1089#  5660
SW10  = 002000  1070#  6334  6477  6615  6969
SW11  = 004000  1069#
SW12  = 010000  1068#
SW13  = 020000  1067#  3767  6480  6497  6529  6618  6727  6793  6936  6972  6990  7326  7409
                7497   7656  7678  7833  7970  8442
SW14  = 040000  1066#  3767  3791  4243  5192  6013  6482  6508  6886  6916  7669  8274
SW15  = 100000  1065#  6506  6850
```

```
SW2    = 000004            1088#
SW3    = 000010            1087#   6531
SW4    = 000020            1086#   6886    6916    7669
SW5    = 000040            1085#   6013
SW6    = 000100            1084#   7980
SW7    = 000200            1083#
SW8    = 000400            1082#
SW9    = 001000            1081#
SYS000  057557             3469   10084#
SYS001  057567             3493   10886#
SYS002  057607             3519   10889#
SYS003  057641             3532   10894#
SYS004  057645             3561   10895#
SYS005  057677             3611   10900#
SYS006  057722             3644   10904#
SYS007  057745             3634   10908#
SYS008  057772             3741   10912#
SYS009  060040             3587   10919#
SYS010  060064             3667   10923#
S.ACLO= 000100             1308#
S.BRHM= 000100             1323#
S.BRKE= 040000             1345#
S.CART= 000400             1325#
S.DCLO= 010000             1315#
S.DIB  = 002000            1341#
S.DOOR= 000200             1324#
S.DRA  = 000040            1294#
S.DROT= 020000             1316#
S.DRY  = 000200            1296#
S.DSC  = 040000            1303#   8818    8975    8992    9062    9079    9094
S.FLT  = 000200            1309#   8970    9075
S.FORM= 001000             1298#
S.FWD  = 002000            1327#
S.HDFL= 000200             1338#
S.HDHM= 000040             1322#
S.ICYL= 000040             1307#
S.ILF  = 000400            1310#
S.LIMD= 020000             1344#
S.LOAD= 010000             1329#
S.MHD  = 000400            1339#
S.NMOV= 010000             1343#
S.OFF  = 002000            1299#
S.PAR  = 001000            1311#   8821    9354
S.PIP  = 020000            1302#   8829    9040
S.PLO  = 004000            1342#
S.REV  = 004000            1328#
S.RTZ  = 020000            1330#
S.SECT= 000020             1335#
S.SKI  = 002000            1312#
S.SPIN= 010000             1301#
S.SPLS= 010000             1314#
S.SPOK= 001000             1326#
S.TYPE= 000400             1297#
S.UNLD= 040000             1331#
S.UNS  = 040000            1317#
S.VV   = 000100            1295#
```

```
S.WCLK= 000040  1336#
S.WGAT= 000100  1337#
S.WLE = 004000  1313#
S.WRL = 004000  1300#
S.XDOK= 000020  1321#
S.XERR= 001000  1340#
S1$     011110  3467    3473    3699#
S2$     011122  3497    3701#
S3$     011152  3536    3705    3708#
S4$     011210  3565    3714    3718#
S5$     011216  3591    3719#
S6$     011222  3615    3720#
S7$     011230  3648    3721#
S8$     011236  3671    3722#
TBITVE= 000014  1123#
TEST    013536  3440    3888    3903    3924    3936    4033    4059    4136#
TESTDV  012036  3861    3881#
TES.1   013612  3436    4154#   4170
TIMHR   001376  1800#   8553*   8559
TIMMIN  001400  1801#   8554*   8568
TIMOUT  025674  6127    6172#
TIMSEC  001402  1802#   8555*   8570
TIM000  065334  8573   11454#
TIM001  065341  8557   11455#
TKVEC = 000060  1130#  10199*  10200*
TMPCNT  001570  1926#   3529*   3530    3543    3556
TMPSEC  001576  1928#   5310*   5311*   5312*   5316    5319*   5322*   5323    5330
TMPTRK  001572  1927#   5296*   5297*   5298*   5302    5305*   5308*   5309    5326
TPVEC = 000064  1131#
TRAPVE= 000034  1129#   3314*   3315*   3347*   3746*  10453*
TRTVEC= 000014  1124#
TSTDEF  020516  3470    3494    3533    3562    3588    3612    3645    3668    4529    4571    4612    4653    4677
                4699    4724    4747    4769    4789    4815    4850    4871    4963#   4983    5044
TSTDRP  037670  7096    7867#
TYPE  = 104401  3343    3469    3483    3485    3493    3512    3514    3519    3532    3551    3553    3561    3578
                3580    3587    3604    3606    3611    3634    3637    3639    3644    3661    3663    3667    3691
                3693    3731    3741    3744    3842    3853    3854    3875    3895    3908    3909    3957    3970
                3971    3993    4006    4007    4026    4040    4052    4064    4065    4079    4080    4090    4093
                4096    4100    4104    4108    4112    4115    4118    4121    4124    4127    4130    4278    4337
                4344    4345    4348    4492    4499    4500    4503    4526    4528    4555    4557    4570    4596
                4598    4611    4637    4639    4652    4666    4668    4676    4690    4692    4697    4714    4716
                4722    4738    4740    4746    4760    4762    4767    4781    4783    4788    4806    4808    4814
                4839    4841    4849    4862    4864    4869    4870    4882    4884    4982    4994    5041    5043
                5141    5143    5360    6208    6211    6220    6223    6228    6231    6234    6237    6240    6243
                6246    6249    6260    6272    6275    6336    6340    6346    6351    6356    6362    6367    6373
                6381    6385    6395    6397    6400    6428    6433    6479    6486    6489    6492    6496    6510
                6511    6539    6545    6547    6554    6557    6586    6608    6617    6622    6630    6633    6634
                6638    6641    6645    6649    6654    6658    6662    6667    6668    6672    6675    6679    6683
                6689    6692    6697    6701    6722    6726    6731    6737    6798    6799    6801    6807    6819
                6822    6825    6828    6831    6834    6837    6888    6918    6928    6938    6971    6976    6978
                6989    7328    7334    7335    7411    7412    7415    7416    7419    7499    7502    7584    7595
                7617    7619    7622    7628    7631    7635    7658    7662    7672    7680    7731    7835    7837
                7842    7845    7848    7853    7854    7901    7904    7910    7915    7920    7925    7930    7933
                7939    7945    7947    7950    7956    7960    7964    7968    7974    8421    8434    8437    8438
                8441    8446    8449    8452    8455    8458    8461    8467    8573    9935    9938    9983   10074
               10138   10144   10149   10153   10158   10159   10161   10164   10168   10212   10223   10230   10231
```

```
                    10234   10246   10257   10274   10455   10628   10740#
T.ASOF  001424       1816#   6431    8782*   8783    8811    8856*   8857    8939*   8940    8947    8983    9024    9032
                     9056    9088    9247*
T.BA    001416       1813#   9021    9243*
T.CONT= 000012       1389#   7851
T.CS1   001410       1808#   6177*   6178    6206    6419    6516*   8752*   8779*   8780    8793    8808*   8841*   8843
                     8849    8878*   8900*   8902    8934    8966*   8980*   9017*   9018    9029*   9053*   9072*   9085*
                     9135*   9136    9176    9180*   9181*   9183    9187    9200    9240*   9287*   9288    9328*   9335*
                     9342*   9349*   9832*   9833    9838    9844*
T.CS2   001412       1810#   6423    8728*   8729    8744    8747    8762    8859*   8870    8906    8920    9019    9177*
                     9178*   9179    9189    9198    9202    9241*   9847    9851
T.DA    001420       1814#   9022    9244*
T.DB    001444       1825#   6210
T.DC    001422       1815#   9023    9248*
T.DLT = 001000       1388#   7332
T.DS    001430       1819#   6207    9026    9245*   9849
T.ER    001426       1818#   6379    6425    8909*   8912    9025    9204    9246*
T.MR1   001432       1820#   9249*
T.MR2   001434       1821#   8818    8829    8974    8991    9040    9062    9079    9094    9250*   9331    9338    9345
                     9352
T.MR3   001436       1822#   8821    8881    8969    9075    9251*   9332    9339    9346    9353    9354
T.NER = 000020       1375#   6865
T.PAT   001442       1824#   9253*
T.POS   001440       1823#   9252*
T.WCR   001414       1812#   9020    9242*
UEXATT= 000010       1443#   7892    7965    8450    8788
UFE   = 000400       1226#   5900    6423    7056    8744    8799    9202    9535
UNLOAD= 000107       1166#   9438
UNRECY  031660       6904    6909#   7222    7428    7464    7513    7995    8130
UNS   = 040000       1254#   5926    7876    7908    7984
UPE   = 020000       1231#   5900    6423    7090    8906    9202
VV    = 000100       1266#   4204
WCE   = 040000       1232#   5900    7453    8906    9202
WLE   = 004000       1251#   5926    7876    8912
WRDATA= 000123       1172#   5257    9493
WRHEAD= 000127       1174#
WRL   = 004000       1270#   4334    4355
WRTCHK= 000131       1175#   4362    4363    4399    4426    4428    5209    5263    6001    7246    7446    8049    8147
                     8237    8381    9489    9868
WRTEPK  012656       3755    3872    4015#
WRTGAT= 040000       1289#
WRTVER  015412       4302    4393#
WV0$    015454       4397    4402#   8276
WV1$    016216       4292    4492#   8281
WV2$    016230       4408    4496#
WV5$    016340       4494    4518#
WV9$    016344       4519#
W.DRV   001544       1908#   3386    6438*   8653*   8777*   8833*   8848*   8861*   8875*   8925*   8999*   9015*   9044*
                     9066*   9100*   9411*   9474*   9477*   9530*   9540*   9550*   9855*   9889*
W.MILI  001470       1858#   3393    8645    9613
W.MIN   001476       1874#   9474    9540    9855
W.MTIM  001466       1857#   3393*   8643*   8645*
W.SEC   001472       1871#   9411    9889
W.TIME  001512       1881#   3392*   6436*   8651    8658*   8778*   8832*   8847*   8860*   8876*   8924*   8998*   9014*
                     9043*   9065*   9099*   9408*   9529*   9539*   9549*   9853*   9886*
W.WCK = 000200       1448#   4255    4283    4285    4364    4396    4398    4409    4430    5211    5264    6003    6005
```

```
                        7248    7448    7476    7721    8051    8149    8239    8241    8256    8367    8390    9421    9482
                        9491    9497    9899
W.WTCH  044270          3763    8638#   9520
W.8SEC  001474          1873#   8833    9044    9477
XXDPMG  056224          3731    4026    4052   10747#
$APTHD  001000          1646    1652#
$ASTAT= ****** U       10686   10701
$ATYC   055736         10657   10659#
$ATY1   055712         10655#
$ATY3   055720          9968   10656#
$ATY4   055730         10658#
$AUTOB  001134          1683#  10228
$BASE   001244          1745#
$BDADR  001122          1678#
$BDDAT  001126          1680#
$BELL   001160          1696#   6336    6479    6617    6971
$B2D    055532         10586   10602#
$CDW1   001250          1747#
$CDW2   001252          1748#
$CHARC  053220          9985*   9995*  10002   10028*  10033#
$CKSWR= ****** U       10743
$CMTAG  001100          1666#   3307    3308
$CM3  = 000000          1696#
$CNTLC  054401         10212   10279#
$CNTLG  053752         10179#  10230
$CNTLU  053745         10153   10178#  10246
$CPUOP  001216          1719#
$CRLF   001165          1698#   4130    4348    4503    6249    6275    o397    6547    6557    6634    6658    6668    6701
                        6726    6801    6825    6837    7415    7419    7635    7662    7837    7848    7947    8438    9938
                        9984   10036   10158   10178   10257
$DB2D   055470          4099    4103    4107   10588#
$DDW0   001254          1749#
$DDW1   001256          1750#
$DDW10  001300          1759#
$DDW11  001302          1760#
$DDW12  001304          1761#
$DDW13  001306          1762#
$DDW14  001310          1763#
$DDW15  001312          1764#
$DDW2   001260          1751#
$DDW3   001262          1752#
$DDW4   001264          1753#
$DDW5   001266          1754#
$DDW6   001270          1755#
$DDW7   001272          1756#
$DDW8   001274          1757#
$DDW9   001276          1758#
$DECNT  055574         10583*  10598*  10618#
$DECVL  055676         10602   10628   10651#
$DEVCT  001200          1710#
$DEVM   001246          1746#
$ENV    001210          1715#   9963   10664   10688
$ENVM   001211          1716#   3335    9965    9970   10666
$ERFLG  001103          1669#
$ERMAX  001115          1675#
$ERRPC  001116          1676#
```

```
$ERRTB  001314          1769#
$ERTT_  001112          1673#
$ETABL  001210          1714#
$ETEND  001314          1658    1767#
$FATAL  001172          1707#   10692*
$FFLG   056156         10655*  10658*  10686   10695*  10703#
$FILLC  001156          1694#   9988    10036
$FILLS  001155          1693#   10036   11550
$GDADR  001120          1677#
$GDDAT  001124          1679#
$GTSWR= ****** U        10742
$HD   = 000000           992
$HIBTS  001000          1653#
$HINUM  055422          3445*   5244    5248    5254    5301    5315    10536   10544   10549*  10554#
$HIOCT  054540         10323*  10336#
$ICNT   001104          1670#
$INTAG  001135          1684#
$ITEMB  001114          1674#
$LF     001166          1699#   10036   10168   10178
$LFLG   056155         10696*  10702#
$LKCSB  001336          1779#   8494    8514*
$LKCSR  001334          1778#   8515*
$LKS    001342          1781#   8500    8519*
$LLVEC  001344          1782#   8518
$LONUM  055424          3446*   5227    5290    10535   10542   10548*  10555#
$LPADR  001106          1671#
$LPERR  001110          1672#
$LPVEC  001340          1780#   8513
$LSCS   001346          1783#
$LSDB   001350          1784#
$LSTAD  055044          3400    3491    3700    10429*  10433#
$MADR1  001222          1732#
$MADR2  001226          1736#
$MADR3  001232          1739#
$MADR4  001236          1742#
$MAIL   001170          1654    1658    1705#   3334    9963
$MAMS1  001220          1726#
$MAMS2  001224          1734#
$MAMS3  001230          1737#
$MAMS4  001234          1740#
$MBADR  001002          1654#
$MFLG   056154         10656*  10662   10697*  10701#
$MNEW   053770         10182#  10234
$MSGAD  001204          1712#  10672*  10675
$MSGLG  001206          1713#  10677*
$MSGTY  001170          1706#  10670   10678*  10690   10694*
$MSWR   053757         10180#  10231
$MTYP1  001221          1727#
$MTYP2  001225          1735#
$MTYP3  001231          1738#
$MTYP4  001235          1741#
$MULT   055210          4440    5074    5111    5124    5230    5291    5304    5318    5328    7815    8313    10484#
$NULL   001154          1692#   9990   10036   11554
$PASS   001176          1709#   3334*
$PASTM  001006          1656#
$POWER  055176         10455   10462#
```

```
$PWRCT  055172          10445* 10446* 10447* 10449* 10461#
$PWRDN  055C46           3316  10439# 10451
$PWRUP  055066          10440  10445#
$QUES   001164           1697#  3485   3514   3553   3580   3606   3639   3663   3693   4557   4598   4639   4668
                         4692   4716   4740   4762   4783   4808   4841   4864   4884   5143  10036  10161  10178
                        10274
$RAND   055324           5226   5285  10531#
$RDCHR  053330          10096# 10743
$RDDEC= ****** U        10745
$RDLIN  053460          10126# 10744
$RDOCT= ****** U        10745
$RDSZ = 000031          10119#
$R2A  = ****** U        10745
$SAVRE= ****** U        10745
$SB2D   055426           4111   4114   4117   4120   4123   4126   4129  10574#
$SETUP= 000014           3292#  3313   3314   3316   3318   3343   3344  10085  10184
$SIZE   054714           3349  10406#
$STUP = 177777           3292#
$SWR  = 162000            979#   992   1011   1012   1013   1014   1015   1696
$SWREG  001212           1717#  3337
$TESTN  001174           1708#
$TKB    001146           1689#  10009  10016  10036  10083  10100  10106  10201  10207  10217  10241
$TKINT  054002           3745  10199#
$TKS    001144           1688#  5216*  5368*  10007  10014  10036  10083  10098  10104  10202* 10213* 10218* 10235*
                        10238  10258*
$TKSRV  054032          10199  10207#
$TN   = 000001            979#   992
$TNPW   055652          10584  10585  10641#
$TNPWR  055626          10599  10600  10631#
$TPB    001152           1691#  10025* 10036  11546* 11549* 11554* 11565* 11584*
$TPFLG  001157           1695#   9957  10036
$TPS    001150           1690#  10023  10036  11544  11547  11552  11563  11582
$TRAP   056160           3314   10717#
$TRAP2  056202          10728# 10739
$TRP  = 000004          10732# 10741# 10743  10744# 10745#
$TRPAD  056214          10722  10739#
$TSTM   001004           1655#
$TSTNM  001102           1668#
$TTYIN  053714          10128  10129  10141  10159  10173  10177#
$TYPBN= ****** U        10741
$TYPDS= ****** U        10741
$TYPE   052670           9957#  10683  10732  10740
$TYPEC  053102           9987   9994   10001  10006# 10260
$TYPEX  053222          10029  10031  10034#
$TYPOC= ****** U        10741
$UNIT   001202           1711#
$UNITM  001010           1657#
$USWR   001214           1718#
$VECT1  001240           1743#
$VECT2  001242           1744#
$XOFF = 000023          10011  10036
$XON  = 000021          10018  10036  10111
$.$.SM= ****** U         1857   3365   3727   8676   8892   9135   9146   9184   9508   9520   9570   9590   9605
                         9632   9921   9933
$2DEC   055536          10603# 10619
.     = 067505           1619#  1623#  1629#  1634#  1636#  1642   1643#  1645#  1647#  1665#  1700   3311   3346#
```

```
                        6402    7337    7856   10036   10077#  10083   10177#  10178   10184#  10442   10651#  10704#  10838#
                       10882#  11458#  11462#  11464   11542   11609#  11610   11611
.$ASTA= ****** U       10656   10659
.$X   = 001000          1642#   1647
..ADD   066353         11513   11604#
..ASCI  066130         11476   11481   11486   11491   11496   11501   11506   11512   11520   11559#
..BLNK  066376         11521   11608#
..CR    066050         11475   11480   11485   11490   11495   11500   11505   11510   11511   11514   11515   11526   11532
                       11533   11544#  11545
..DUMP  065460          1631   11465#
..HIGH  000226          1633#  11530
..LOW   000224          1632#  11464   11474
..OCT   066154         11479   11484   11489   11494   11499   11504   11509   11518   11523   11570#
..R0    066236         11477   11590#
..R1    066251         11482   11592#
..R2    066264         11487   11594#
..R3    066277         11492   11596#
..R4    066312         11497   11598#
..R5    066325         11502   11600#
..SP    066340         11507   11602#
..STCK  065456         11463#  11466*  11467   11508   11540
```

```
COMMEN      1#   1133#
DEFLT    1617#   3470    3494    3533    3562    3588    3612    3645    3668    4529    4571    4612    4653    4677    4699
         4724    4747    4769    4789    4815    4850    4871    5044
ENDCOM      1#   1133#
ERROR    1027#
ESCAPE      1#   1133#
GETPRI      1#   1133#  10416
GETSWR      1#   1133#   3344#
MULT        1#   1133#
M.DUMP      2#  11459
NEWTST      1#   1133#
PARMBK      2#   1969    2062    2155    2248    2341    2434    2527    2620    2714
PARMDF      2#   1390
PARMS    1617#   2001    2094    2187    2280    2373    2466    2559    2652
POP         1#   1133#  10070   10511   10550   10621   10698   10699   11534
PUSH        1#   1133#  10050   10484   10531   10574   10588   10659   10661   10682   11468
P.DRVR      2#   1805
QINPUT   1617#   3481    3510    3549    3576    3602    3635    3659    3689    4553    4594    4635    4664    4688    4712
         4736    4758    4779    4804    4837    4860    4880
REGDEF      2#   1139
REPORT      1#   1133#
R.DRIV      2#   8596
R.QDRV      2#   9641
SCANST   1617#   4543    4584    4625    5059    5079    5096
SCOPE    1028#
SETPRI      1#   1133#
SETTRA  10732#  10743   10744
SETUP       1#   1133#   3306
SKIP        1#   1133#
SLASH       1#   1133#
SPACE    1133#
STARS       1#    993    1006   1133#   1639    1641    1648    1661    1700    1703    3815    3839    4948    4961    5007
         5037    5151    5186    5393    5410    5471    5489    5564    5594    5678    5699    5864    5885    6705    6715
         6755    6769    8605    8636    8679    8719    9115    9132    9150    9174    9225    9238    9257    9283    9302
         9324    9367    9394    9643    9652    9677    9690    9711    9724    9750    9761    9803    9817    9942   10039
        10048   10082   10088   10119   10188   10197   10283   10296   10339   10351   10400   10434   10468   10482   10522
        10559   10572   10654   10711
SWRSU       1#   1133#   3318#
S.DRVE      2#   1291
TRMTRP  10732#
TYPBIN      1#   1133#
TYPDEC      1#   1133#
TYPNAM      1#   1133#   3339
TYPNUM      1#   1133#
TYPOCS      1#   1133#
TYPOCT      1#   1133#
TYPTXT      1#   1133#
$DECBN      2#  10337
$OCTBN      2#  10281
$$CMRE   1659#
$$CMTM   1659#
$$ESCA      1#   1133#
$$NEWT      1#   1133#
$$SET   10732#  10743   10744
$$SETM   3334#
$$SKIP      1#   1133#
```

```
.EQUAT      1#    979#   1023
.HEADE      1#    979#    982
.KT11       1#
.SETUP      1#    979#   3292
.SWRHI      1#    979#   1007
.SWRLO    979#   1015#   1017    1019    1021    1022
.$ACT1      1#
.$APTB      1#   1701#
.$APTH      1#    979#   1637
.$APTY      1#    979#  10652
.$ASTA      1#
.$CATC      1#    979#   1617
.$CMTA      1#    979#   1659
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#    979#  10080
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#    979#  10398
.$SUPR      1#
.$TRAP      1#    979#  10709
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#    979#   9940
.$TYPO      1#
.$40CA      1#
.1170       1#

. ABS.  067505     000

 ERRORS DETECTED:  0

 CZR6PD,CZR6PD/SOL/CRF/NL:TOC/EQ:QNEWSW=SYSMAC.SML,DRIV11.P11,DUMP.P11,CZR6PD.P11
 RUN-TIME: 27 34 2 SECONDS
 RUN-TIME RATIO: 118/64=1.8
 CORE USED:  59K  (118 PAGES)
```