RK05/RK11

UTILITY PACKAGE

MD-11-DZRKI-C

EP-DZRKI-C-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRKI-C-D
PRODUCT NAME:	RK11 UTILITY PACKAGE
DATE CREATED:	MARCH 1, 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	BOB COLLINS
REVISED BY:	JIM KAPADIA TOM SAWYER

TABLE OF CONTENTS

1.0 ABSTARCT

2.0 REQUIREMENTS
2.1 EQUIPMENT
2.2 STORAGE
2.3 PRELIMINARY PROGRAMS

3.0 LOADING PROCEDURE

4.0 STARTING PROCEDURE

5.0 OPERATING PROCEDURE

6.0 ERRORS

7.0 RESTRICTIONS

8.0 EXECUTION TIME

9.0 PROGRAM DESCRIPTION
9.1 PROGRAM INDEX
9.2 COMPATIBILITY PACKAGE
9.3 OSCILLATING SEEK PACKAGE
9.4 FORMATTER SURFACE VERIFIER
9.5 RK05 CONTROL PANEL TEST
9.6 RK05 CONTROL PANEL TEST # 2
9.7 HEAD ALIGNMENT ROUTINE
9.8 (DISK) POWER FAILURE TEST
9.9 SECTION SPECIAL
9.10 COMPATIBILITY ERROR RECOVERY

1. ABSTRACT

1.1 THIS PACKAGE CONTAINS 4 INDIVIDUAL UTILIT PROGRAMS FOR THE RKXX PLUS A MINI-MONITOR WHICH ALLOWS TEST SELECTION AND PARAMETER INPUT VIA THE CONSOLE DEVICE. ALL UTILITY PACKAGES ARE EXPLAINED IN DETAIL IN PARAGRAPH 8.

2. REQUIREMENTS

2.1 EQUIPMENT
PDP-11 WITH RKXX DRIVE(S)

2.2 STORAGE
THIS PROGRAM REQUIRES 8K

2.3 PRELIMINARY PROGRAMS
THIS IS NOT A DIAGNOSTIC, PACKAGE IT IS ASSUMED THAT ALL EQUIPMENT IS FUNCTIONAL

3. LOADING PROCEDURE

3.1 METHOD
PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED
A. ABSOLUTE LOADER MUST BE IN MEMORY.
B. PLACE BINARY TAPE IN READER.
C. LOAD ADDRESS *7500 (*DETERMINED BY LOCATION OF LOADER).
D. PRESS "START" PROGRAM WILL LOAD.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS
NONE

4.2 STARTING ADDRESS
200 MINI MONITOR

4.3 PROGRAM AND/OR OPERATOR ACTION
LOAD PROGRAM INTO MEMORY
SET SWITCH REGISTER TO STARTING ADDRESS (200)
LOAD ADDRESS
PRESS START

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11 34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' WHENEVER THE PROGRAM ENTERS THE SCOPE ROUTINE OR BEGINS A NEW TEST. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE

TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

EACH TIME SWITCH SETTING ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT REQUIRED. 'RJBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

PROGRAM WILL TYPE MINI MONITOR ROUTINE

5. OPERATING PROCED'URE

5.1 OPERATIONAL SWITCH SETTINGS

SEE SEC. 9.0 FOR SWITCHES APPLICABLE TO INDIVIDUAL ROUTINES.

5.2 SUBROUTINE ABSTRACTS

NOT APPLICABLE

5.3 PROGRAM AND/OR OPERATOR ACTOR

SEE INDIVIDUAL PACKAGE DESCRIPTION (PARAGRAPH 9)

6. ERRORS

6.1 ERROR HALTS AND DESCRIPTION

IF HALTED A MAJOR PROBLEM EXIST CHECK CODE AT HALT PC TO DETERMINE WHAT OCCURRED.

6.2 ERROR RECOVERY

EXPLAINED IN DETAIL IN INDIVIDUAL PACKAGE DESCRIPTION (PARAGRAPH 9)

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

IT IS NOT RECOMMENDED THAT YOU START AT AN ADDRESS OTHER THAN 200, (REASON EXPLAINED IN PARAGRAPH 9.1) UNLESS DIRECTED TO BY THE PROGRAM.

7.2 OPERATIONAL RESTRICTIONS

EXPLAINED IN DETAIL IN INDIVIDUAL PACKAGE DESCRIPTIONS (PARAGRAPH. 9)

8. EXECUTION TIME

VARIES WITH SELECTED ROUTINE, NUMBER OF DRIVES, ETC.

9. PROGRAM DESCRIPTION

THE RK11 UTILITY PACKAGE IS DIVIDED INTO EIGHT SECTIONS
WHICH ALLOW COMPATABILITY TESTING, OSCILLATING SEEKS FOR SERVO
ADJUSTMENT AND SEEK LOGIC WAVEFORM ANALYSIS, PACK FORMATTING
AND SURFACE VERIFICATION, AND FRONT PANEL TESTING (INDICATOR LAMPS,
SWITCHES, INTERLOCKS, ETC) AND VERIFICATION.
THE PACKAGE IS DIVIDED INTO FIVE SECTIONS

SECTION	NAME
0	INDEX
1	COMPATIBILITY TEST
2	OSCILLATING SEEK PACKAGE
3	FORMATTER SURFACE VERIFIER
4	FRONT PANEL TEST
5	RK05 CONTROL PANEL TEST #2
6	HEAD ALIGNMENT ROUTINE
7	POWER FAILURE (DURING WRITE) TEST

NOTE: NORMAL LINKAGE TO ANY OF THESE PACKAGES IS THRU
SECTION 0 (SEE PARAGRAPH 9.1)

9.1 SECTION 0 INDEX

PURPOSE: TO ALLOW THE USER TO SELECT AND RUN TESTS VIA
THE CONSOLE DEVICE IN AN EFFORT TO FREE HIM
FROM REMEMBERING VARIOUS SWITCH SETTINGS.

DESCRIPTION: LOAD START ADDRESS 200, A TABLE IS PRODUCED
WHICH TELLS THE USER THE NAME AND TYPE OF THE
TEST. (TYPE IS AN OCTAL CODE BY WHICH THE USER SELECTS
THE TEST). AFTER THE TABLE IS TYPED, THE QUESTION "TYPE ="
IS ASKED, THE USER THEN TYPES THE NUMERAL 0-4
TO SELECT A TEST.

USE: THIS IS EXAMPLE OF THE ACTUAL OUTPUT:

```
RK11 UTILITY PACKAGE

NAME                              TYPE
INDEX                              0
COMPATIBILITY PACKAGE              1
OSCILLATING SEEK PACKAGE           2
FORMATTER-SURFACE VERIFIER         3
RK05 CONTROL PANEL TEST            4
RK05 CONTROL PANEL TEST #2         5
HEAD ALIGNMENT ROUTINE             6
POWER FAILURE (WRITE) TEST         7

TYPE=X
```

WERE "X" IS THE RESPONSE (0-7) BY THE USER

```
                        ERROR INFO:  ANY ILLEGAL INPUT IS HANDLED, A QUESTION MARK
                                     IS TYPED AND THE QUESTION "TYPE =" IS RE-ASKED.

                    9.2  SECTION 1  COMPATIBILITY PACKAGE

                        PURPOSE:     TO CONFIRM THE FACT THAT A GROUP OF DRIVES
                                     (A MAXIMUM OF EIGHT) ARE TRULY COMPATIBLE.
                                       THIS PACKAGE DOES NOT APPLY TO RK-05F DRIVES.
                        DESCRIPTION: THIS PACKAGE ALLOWS A USER TO AUTOMATICALLY TEST
                                     COMPATIBILITY OF UP TO EIGHT (8) DRIVES SIMPLY
                                     BY STATING THE DRIVE NUMBERS TO BE TESTED.  THE
                                     TEST DOES THE REST, INSTRUCTING THE USER WHERE
                                     TO PLACE THE PACK.  THE LIMITATIONS OF TESTING ARE
                                     IF THERE ARE (2) TWO PROCESSORS, FROM ONE (1) TO
                                     SEVEN (7) DRIVES MAY BE ON SYSTEM ONE, AND
                                     ONLY ONE (1) DRIVE (ANY DRIVE NUMER) MAY
                                     BE ON SYSTEM TWO.
                                     COMPATIBILITY-A DEFINITION, COMPATIBILITY INFERS
                                     MORE THAN THE FACT THAT INFORMATION WHICH
                                     WAS WRITTEN ON ONE DRIVE CAN BE
                                     READ ON ANOTHER.
                                     FOR DRIVES TO BE CONSIDERED TRULY COMPATIBLE
                                     ANY DRIVE SHOULD BE ABLE TO READ WHAT WAS
                                     WRITTEN BY ANY OTHER DRIVE AND ALSO MUST BE ABLE
                                     TO OVERWRITE A PORTION OF INFORMATION WRITTEN
                                     BY ANOTHER DRIVE, WITH NEW INFORMATION, AND
                                     READ IT BACK.  THIS IS A VERY BROAD DEFINITION
                                     BUT IS THE BASIC PREMISE OF TRUE COMPATIBILITY.
                        USE:         THE BELOW IS AN EXAMPLE OF ACTUAL OUTPUT, THE USER
                                     WANTS TO RUN SINGLE PROCESSOR MODE AND TEST
                                     COMPATIBILITY ON THREE (3) DRIVES WHOSE UNIT NUMBERS
                                     ARE 0,1,3.....
                                                **********

                        EXAMPLE 1

                                NAME                          TYPE
                            INDEX                              0
                            COMPATIBILITY PACKAGE              1
                            OSCILLATING SEEK PACKAGE           2
                            FORMATTER-SURFACE VERIFIER         3
                            RK05 CONTROL PANEL TEST            4
                            RK05 CONTROL PANEL TEST #2         5
                            HEAD ALIGNMENT ROUTINE             6
                            POWER FAILURE (WRITE) TEST         7

                            TYPE=1
                            DRIVE NUMBERS ON SYSTEM 1=0,1,3.

                            IS THERE A SECOND SYSTEM?N
                            MOUNT PACK ON DRIVE #0
                            MAKE PACK WRITE ENABLE
                            PRESS CONTINUE WHEN DRIVE RDY
                            MOUNT PACK ON DRIVE #1
                            MAKE PACK WRITE ENABLE
                            PRESS CONTINUE WHEN DRIVE RDY
```

```
                        MOUNT PACK ON DRIVE #3
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE RDY
                        MOUNT PACK ON DRIVE #0
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE RDY
                        MOUNT PACK ON DRIVE #1
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE RDY
                        MOUNT PACK ON DRIVE #3
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE RDY
                        DONE!
                                RK11 UTILITY PACKAGE

                                NAME                    TYPE
                        INDEX                            0
                        COMPATIBILITY PACKAGE            1
                                                **********

                                ...THE USER SELECTED TYPE ONE (1) AND RECEIVED
                        THE MESSAGE RKXX COMPATIBILITY PACKAGE AND WAS THEN
                        ASKED FOR SYSTEM 1 DRIVES HE TYPES EACH SELECTED
                        DRIVE NUMBER SEPARATED BY COMMAS HE TERMINATES THE
                        STRING WITH A PERIOD THEN A CARRIAGE RETURN
                        HE IS ASKED IF THERE IS A SECOND SYSTEM,
                        HE TYPES N FOR NO.
                        HE NOW RECEIVES A STRING OF MOVE DIRECTIVES TELLING
                        HIM EXACTLY WHERE TO MOVE THE TEST PACK AND WHAT TO
                        DO.  FINALLY THE USER RECEIVES THE MESSAGE "DONE!"
                        INDICATING A SUCCESSFUL PASS.
                        AT THIS POINT ANY DRIVE WHICH HAS NOT BEEN
                        DECLARED DOWN AND DID NOT RECEIVE AN ERROR*
                        MESSAGE IS COMPATIBLE WITH ANY OTHER SELECTED
                        DRIVE MEETING THE SAME CONDITIONS.
                        FINALLY THE INDEX ROUTINE IS AUTOMATICALLY
                        RE-ENTERED AND USER IS READY TO MAKE ANOTHER
                        SELECTION.
                        *SEE ERROR INFO TO DETERMINE THE TYPE OF ERROR
                        WHICH CONSTITUTES INCOMPATABILITY.
                                                **********

                EXAMPLE 2

                                THE USER NOW DESIRES TO TEST COMPATIBILITY ON
                                TWO SYSTEMS HE HAS UNITS 0,1 ON SYSTEM ONE
                                AND UNIT 0 ON SYSTEM 2. IT GOES LIKE THIS....
                                RK11 UTILITY PACKAGE

                                NAME                    TYPE
                        INDEX                            0
                        COMPATIBILITY PACKAGE            1
                        OSCILLATING SEEK PACKAGE         2
                        FORMATTER-SURFACE VERIFIER       3
                        RK05 CONTROL PANEL TEST          4
                        RK05 CONTROL PANEL TEST #2       5
```

```
                                        HEAD ALIGNMENT ROUTINE          6
                                        POWER FAILURE (WRITE) TEST      7


                                        TYPE=1
                                        DRIVE NUMBERS ON SYSTEM 1=1,0

                                        IS THERE A SECOND SYSTEM?Y
                                        DRIVE # =0
                                        MOUNT PACK ON DRIVE #1
                                        MAKE PACK WRITE ENABLE
                                        PRESS CONTINUE WHEN DRIVE RDY
                                        MOUNT PACK ON DRIVE #0
                                        MAKE PACK WRITE ENABLE
                                        PRESS CONTINUE WHEN DRIVE RDY
                                        LOAD AND START ADDRESS 210 ON SYSTEM #2
                                        AND TYPE THE BELOW WHEN ASKED ON SYSTEM #2
                                        AND TYPE THE BELOW WHEN ASKED FOR IT.
                                        WORD            1=000002
                                        WORD            2=000200

                                                          **********
                                                ...THE ONLY DIFFERENCE BETWEEN THIS AND SINGLE
                                        SYSTEM IS THE NEW DIRECTIVE TO LOAD START 210
                                        ETC.  THE USER NOW LOADS AND STARTS SYSTEM TWO
                                        AND THE BELOW IS TYPED...
                                                          **********

                                        COMPATIBILITY-SYSTEM#2
                                        WORD            1=000002

                                        WORD            2=000200

                                        MOUNT PACK ON DRIVE #0
                                        MAKE PACK WRITE ENABLE
                                        PRESS CONTINUE WHEN DRIVE RDY
                                        DONE SYSTEM 2 RESTART SYSTEM 1, TYPE WORD 000077

                                                          **********
                                                .....THE USER RESPONSE TO THE QUESTION WORD 1 =
                                        BY TYPING WORD 1 FROM PROCESSOR ONE AND
                                        WORD 2 =, BY TYPING WORD TWO FROM PROCESSOR 1
                                        HE RECEIVES THE MOUNT COMMAND MOVES THE TEST PACK
                                        TO SYSTEM TWO, DRIVE NUMBER (0), AND PRESSES
                                        CONTINUE.  NOW THE MESSAGE TO RETURN TO SYSTEM
                                        ONE*

                                        *SYSTEM ONE HAS BEEN IN A HALT STATE AND
                                        SHOULD BE LEFT THAT WAY UNTIL THE RETURN FROM
                                        SYSTEM TWO SO THAT TABLES, ETC. BUILT FOR THE
                                        TEST WILL NOT BE DISTURBED.
                                                          **********

                                        WORD=000077

                                        MOUNT PACK ON DRIVE #1
```

```
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE RDY
                        MOUNT PACK ON DRIVE #0
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE RDY
                        DONE!
                                RK11 UTILITY PACKAGE

                                NAME                        TYPE
                        INDEX                                0
                        COMPATIBILITY PACKAGE                1
                        OSCILLATING SEEK PACKAGE             2
                        FORMATTER-SURFACE VERIFIER           3
                        RK05 CONTROL PANEL TEST              4
                        RK05 CONTROL PANEL TEST #2           5
                        HEAD ALIGNMENT ROUTINE               6
                        POWER FAILURE (WRITE) TEST           7

                        TYPE=

                                                  **********
                        THE USER NOW PRESSES CONTINUE ON PROCESSOR ONE
                        AND IN RESPONSE TO THE QUESTION, WORD =, TYPES
                        THE WORD GIVEN TO HIM FROM PROCESSOR TWO
                        THEN EVERYTHING BECOMES THE SAME AS A SINGLE
                        SYSTEM.  THE USER MEARLY FOLLOWS DIRECTIONS.
                    ERROR INFO: SEE PARAGRAPH 9.6 SPECIAL SECTION

                9.3 SECTION 2  OSCILLATING SEEK PACKAGE

                    PURPOSE:        TO ALLOW THE USER TO MAKE SERVO ADJUSTMENTS
                                    AND/OR SEEK LOGIC CHECKOUT BY PERFORMING
                                    SEEKS BETWEEN USER SPECIFIED ADDRESS
                    DESCRIPTION:    SELECT TYPE 2, THE USER THEN INSERTS
                                    THE DRIVES TO BE TESTED IN SW0 TO SW7
                                    OF THE SWITCH REGISTER.  A SWITCH IS SET
                                    FOR EACH DRIVE (E.G. SW2 TO TEST DRIVE 2.
                                     THE USER THEN INSERTS THE
                                    ADDRESS TO SEEK IN THE SWR. IF BOTH ADDRESS
                                    ARE LEGAL, 50 CYCLES (100 SEEKS) WILL BE
                                    MADE BETWEEN THE SPECIFIED ADDRESS THEN
                                    THE PROGRAM WILL LOOK AT THE SWR FOR
                                    POSSIBLE CHANGES THIS SHOULD ALLOW FOR GOOD
                                    STABLE TRACES ON AN OSCILLISCOPE.
                                    IT SHOULD BE NOTED THAT THE OSCILLATING
                                    SEEKS BETWEEN THE SPECIFIED CYLINDERS
                                    ARE DONE ON ALL AVAILABLE DRIVES.
                                    THE ONLY WAY TO EXIT IS HALT!, LOAD ADDRESS 200,
                                    HIT START.
                    USE:            SELECT TYPE 2, RESPOND TO QUESTION WITH UNIT NUMBER...
                                    TYPE=2
                                    OSCILLATING SEEK PACKAGE
                                    SET SW0 TO SW7 TO SELECT THE DRIVES TO TEST
                                    AND CONTINUE.  IF ALL SWITCHES ARE RESET,
                                    ALL AVAILABLE DRIVES WILL BE TESTED.
                                    TOGGLE THE "FIRST CYLINDER ADDRESS" (OUTER LIMIT)
```

```
                                        INTO THE LOW BYTE (BITO-7) OF THE SWITCH REGISTER AND THE "LAS
                                        CYLINDER ADDRESS" (INNER LIMIT) INTO THE HIGH
                                        BYTE (BIT8-15), THEN PRESS CONTINUE
                                        ...FOLLOW INSTRUCTIONS TYPED
                              ERROR INFO: IF AN ILLEGAL ADDRESS IS SELECTED A MESSAGE IS TYPED
                                        AND USER MEARLY SELECTS LEGAL ADDRESS AND DEPRESSES
                                        CONTINUE
                                        EXAMPLE TYPEOUT
                                        INVALID ADDRESS IN SWITCH REGISTER TRY AGAIN
                                        INVALID ADDRESS IN SWITCH REGISTER TRY AGAIN
                                        INVALID ADDRESS IN SWITCH REGISTER TRY AGAIN

                          **NOTE:**     BOTH DRIVES OF AN RK-05F SHOULD NOT BE SELECTED
                                        FOR TESTING AT THE SAME TIME.

                          9.4  SECTION 3  FORMATTER-SURFACE VERIFIER

                               PURPOSE:     TO FORMAT VIRGIN PACKS OR REFORMAT AN OLDER
                                            PACK AND VERIFY ITS SURFACE
                               DESCRIPTION: SELECT TYPE 3, RESPOND TO THE QUESTION
                                            BY SETTING SWITCHES CORRESPONDING TO
                                            DRIVE NUMBERS TO BE FORMATTED.  THUS IF
                                            DRIVES 0,1,2 ARE TO BE FORMATTED SET
                                            SWITCHES 0,1,2.  THE DRIVES ARE FORMATTED
                                            ONE AFTER ANOTHER AT COMPLETION PACK GOOD
                                            MESSAGE IS TYPED AND PACK IS FORMATTED.
                               USE:         SELECT TYPE 3, RESPOND TO QUESTION WITH
                                            SETTING OF SWITCH REGISTER.
                                                      **********

                                    RK11 UTILITY PACKAGE

                                    NAME                    TYPE
                              INDEX                          0
                              COMPATIBILITY PACKAGE          1

                              OSCILLATING SEEK PACKAGE       2
                              FORMATTER-SURFACE VERIFIER     3
                              RK05 CONTROL PANEL TEST        4
                              RK05 CONTROL PANEL TEST #2     5
                              HEAD ALIGNMENT ROUTINE         6
                              POWER FAILURE (WRITE) TEST     7

                              TYPE=3
                              FORMATTER-SURFACE VERIFIER, SET SW REG WITH DRIVE #'S

                              PACK GOOD.
                                    RK11 UTILITY PACKAGE

                                    NAME                    TYPE
                              INDEX                          0
                              COMPATIBILITY PACKAGE          1
                                                    **********
                              AFTER THE PACK IS FORMATTED A GOOD MESSAGE IS
                              GIVEN AND A CHECK IS MADE TO SEE IF THERE ARE
                              ANY MORE PACKS TO BE FORMATTED.  IF THERE ARE
```

```
                        NONE CONTROL IS TRANSFERRED TO THE MINI-MONITOR
                    ERROR INFO: DRIVE PROBLEM, IF THE MESSAGE....
                                SYSTEM ERROR
                                ....IS TYPED IT INDICATES A FAULTY DRIVE OR
                                CONTROLLER, RUN DIAGNOSTICS, THE PROCESSOR WILL HALT
                                PRESS CONTINUE TO RETURN TO MINI MONITOR.
                                BAD SPOT, OR SURFACE PROBLEM, ETC.

                                PACK FAILED AT (IN OCTAL) CYLINDER SECTOR SURFACE

            9.5  SECTION 4      RK05 CONTROL PANEL TEST

                 PURPOSE:       TO INSURE ALL SWITCHES INDICATOR LAMPS, AND INTERLOCKS
                                ARE FUNCTIONAL IN THE RK05
                 DESCRIPTION:   SELECT TYPE 4, RESPOND TO QUESTION WITH UNIT NUMBER, FOLLOW
                                DIRECTIONS GIVEN.  AT COMPLETION MESSAGE "DONE!" IS GIVEN
                 USE:           SELECT TYPE 4, RESPOND TO QUESTION WITH THE UNIT NUMBER....
                                           **********
                                NAME                    TYPE
                            INDEX                       0
                            COMPATABILITY PACKAGE       1
                            OSCILLATING SEEK PACKAGE    2
                            FORMATTER-SURFACE VERIFIER  3
                            RK05 CONTROL PANEL TEST     4
                            RK05 CONTROL PANEL TEST #2  5
                            HEAD ALIGNMENT ROUTINE      6
                            POWER FAILURE (WRITE) TEST  7

                            TYPE=4
                            RK05 CONTROL PANEL TEST, WHICH DRIVE?0
                            MOUNT PACK ON DRIVE#0
                            PLACE DRIVE IN RUN ;SHOULD SEE THE RUN,
                            POWER, AND ON CYLINDER LAMPS LIGHT.
                            MAKE DRIVE WRITE ENABLE PRESS CONTINUE

                            WRITE PROTECT THE DRIVE THEN PRESS CONTINUE

                            CLEAR WRITE PROTECT THEN PRESS CONTINUE

                            CAUTION! TRY TO OPEN THE DOOR, DO NOT FORCE:
                            DOOR SHOULD NOT OPEN!
                            PRESS CONTINUE WHEN FINISHED

                            PUT DRIVE IN LOAD, WAIT FOR LOAD LIGHT
                            PRESS CONTINUE WHEN FINISHED

                            OPEN THE DOOR, PUT DRIVE IN RUN
                            CAUTION! IF RUN LIGHT ON ERROR! DEPRESS
                            LOAD IMMEDIATELY, CONTINUE WHEN FINISHED

                            REMOVE THE PACK, CLOSE THE DOOR
                            PUT DRIVE IN RUN, DRIVE SHOULD NOT
                            RUN...INTERLOCKS HAVE BEEN CHECKED
                            DONE!
```

```
                                        NAME                    TYPE
                                INDEX                           0
                                COMPATABILITY PACKAGE           1
                                OSCILLATING SEEK PACKAGE        2
                                FORMATTER-SURFACE VERIFIER      3
                                RK05 CONTROL PANEL TEST         4
                                RK05 CONTROL PANEL TEST #2      5
                                HEAD ALIGNMENT ROUTINE          6
                                POWER FAILURE (WRITE) TEST      7

                                TYPE=
                                                        **********

                        9.6  SECTION 5    RK05 CONTROL PANEL TEST #2

                             PURPOSE:     TO GIVE A CONTINUOUS MONITORING AND
                                          CHECKING CAPABILITY FOR THE FOLLOWING
                                          CONDITIONS ON THE VARIOUS DRIVES:
                                          OFF LINE (RDY CLR)/ON LINE (RDY SET)
                                          WRITE PROTECTED/WRITE ENABLED
                                          POWER LOW/POWER UP
                                          SEEK INCOMPLETE/SEEK OK
                             DESCRIPTION: SELECT TYPE 5, PUT ALL THE DRIVES THAT
                                          ARE TO BE MONITORED AND CHECKED ON 'RUN'.
                                          NOTE THAT THIS IS IMPORTANT BECAUSE THE
                                          PROGRAM HAS TO KNOW WHICH DRIVES ARE TO
                                          BE CHECKED.
                             USE:         AFTER HAVING SELECTED TYPE 5 AND PUTTING
                                          THE DRIVES THAT ARE TO BE MONITORED ON
                                          'RUN', THE PROGRAM PRINTS OUT ALL THE
                                          DRIVES THAT ARE 'ON LINE'.

                                          DRIVE 0 ON LINE
                                          DRIVE 1 ON LINE
                                          DRIVE 2 ON LINE

                                          THE PROGRAM, THEN STARTS SCANNING ALL
                                          DRIVES, ONE AFTER THE OTHER. CHECKS IF
                                          THE DRIVE IS ON LINE OR OFF LINE (DRY
                                          SET OR CLEAR). THEN IT CHECKS IF THE
                                          DRIVE IS WRITE ENABLED OR WRITE PROTECTED.
                                          THEN A SEEK (TO CYLINDER 1) IS DONE AND
                                          'DPL' BIT IS CHECKED TO SEE IF DRIVE
                                          POWER IS LOW OR OK. IF THE DRIVE IS
                                          POWERED, IT IS CHECKED IF THE SEEK IS
                                          DONE OR SEEK INCOMPLETE OCCURS. WHEN
                                          EVER ANY CHANGE IN THE STATUS IS FOUND,
                                          IT IS REPORTED. IF THE DRIVEIS PUT ON
                                          'LOAD' AND BACK TO 'RUN', THE PROGRAM
                                          CHECKS IF THE DRIVE COMES ON LINE IN THE
                                          WRITE ENABLED MODE. IF NOT, AN ERROR MESSAGE
                                          (ERROR, NOT WRITE ENABLED) IS REPORTED.
                                          THEN THE DRIVE IS WRITE PROTECTED.
                                          EX: IN A SYSTEM UNDER TEST, IF A DRIVE IS PUT
                                          ON 'LOAD' BY THE USER IT GETS REPORTED.
```

```
                                        IF THE USER SET 'WRITE PROT' IT GETS
                                        REPORTED. THE MESSAGES APPEAR AS FOLLOWING:

                                        DRIVE 0 OFF LINE
                                        DRIVE 1 WRITE PROTECTED
                                        DRIVE 2 SIN
                                        DRIVE 1 WRITE ENABLED
                                        DRIVE 0 POWER LO
                                        DRIVE 2 SEEK OK
                                        DRIVE 0 POWER OK

                                        NOTE THAT ONLY CHANGES IN STATUS ARE
                                        REPORTED. THESE CHANGES HAVE TO BE AFFECTED
                                        BY THE USER, IF ANY CHANGE IN STATUS IS NOT
                                        DETECTED AND REPORTED BY THE PROGRAM
                                        IT MIGHT IMPLY AN ERROR CONDITION.

                9.7  SECTION 6          HEAD ALIGNMENT ROUTINE

                     PURPOSE:           TO PROVIDE A FACILITY FOR HEAD ALIGNMENT,
                                        WITH DYNAMIC SELECTION OF THE UPPER OR
                                        LOWER HEAD.
                     DESCRIPTION:       WHEN THE ROUTINE IS SELECTED THE
                                        FOLLOWING MESSAGE APPEARS:
                                        SET SW0=0 FOR SURFACE 0, SW0=1 FOR SURFACE 1.
                                        SET SW1=1 TO TEST CYL 64, SET SW1=0 TO TEST
                                        CYLINDER 105.
                                        SW2-15=0
                                        PUT ANY SW FROM 2-15 HI TO SELECT NEW DRIVE

                                        THEN THE FOLLOWING QUESTION IS ASKED:
                                              DRIVE?                    THE USER
                                        SHOULD TYPE IN THE DRIVE NUMBER THAT
                                        HE WANTS TO SELECT.  THE DRIVE NUMBER IS
                                        SUFFIXED WITH AN 'F' TO TEST RK-05F TYPE
                                        DRIVES.


                                        TYPE=6
                                        DRIVE=0<CR>

                                        THE UPPER OR THE LOWER HEAD CAN BE
                                        SELECTED BY SWITCH 0. IF SURFACE 0 IS
                                        TO BE SELECTED, PUT SW 0 TO 0. IF
                                        SURFACE 1 IS TO BE SELECTED PUT SW 0 ON 1.
                                        THE HEADS MAY BE POSITONED AT CYLINDER 64
                                        OR CYLINDER 105.  SET SW1=0 FOR CYLINDER 105.
                                        SW1=1 FOR CYLINDER 64.
                                        THE PROGRAM POSITIONS THE HEADS ON THE SELECTED
                                        CYLINDER AND CONTINUOUSLY READS FROM
                                        THE SURFACE SELECTED. IF THE USER WISHES
                                        TO SELECT THE OTHER HEAD OR CYLINDER IT CAN BE
                                        DYNAMICALLY DONE BY FLIPPING SW 0 OR SW 1.
                                        IF SOME OTHER DRIVE IS TO BE SELECTED.
                                        ANY SWITCH BETWEEN SW 2 AND SW 15 SHOULD
                                        BE PUT UP. THE QUESTION - DRIVE' IS
```

ASKED AGAIN. THIS IS A CONTINUOUS ROUTINE,
HENCE TO EXIT A HALT HAS TO BE DONE.

NOTE ALIGNMENT IS DONE WITH AN RK-05J CARTRIDGE
SO IF AN F TYPE DRIVE IS SELECTED, CYLINDER
64 OF THE RK-05J IS CYLINDER 130 OF THE F DRIVE
(EVEN DRIVE). CYLINDER 105 BECOMES CYLINDER 5
OF THE ODD DRIVE ON THE RK-05F.

9.8 SECTION 7 (DISK) POWER FAILURE (DURING WRITE) TEST

PURPOSE: THIS TEST CHECKS THAT DATA WRITTEN ON THE DISK IS NOT DESTROYED WHEN THE DISK SENSES A LOSS OF POWER (POWER FAILS) WHILE DOING A WRITE.

DESCRIPTION: UPON SELECTING THIS TEST, THE PROGRAM FINDS OUT THE FIRST AVAILABLE DRIVE AND INDICATES IT TO THE USER BY TYPING A MESSAGE:

DRIVE X X=DRIVE NUMBER 0,1,...7

THEN IT PROCEEDS TO TO WRITE UNIQUE PATTERNS ON CYLINDERS 0 TO 15 (DECIMAL) OF THAT DRIVE. THE HEADS ARE THEN POSITIONED ON CYLINDER 10 AND THE USER IS ASKED TO DROP POWER ON THAT DRIVE:

DROP POWER

MEANWHILE WRITE IS BEING DONE ON CYLINDER 10. ON GETTING THE ABOVE MESSAGE THE USER SHOULD DROP THE POWER ON THAT DRIVE. ON SENSING A LOSS OF POWER, THE PROGRAM WILL ASK THE USER TO PUT THE POWER ON AGAIN:

POWER ON

ON RECEIVING THE ABOVE MESSAGE THE USER SHOULD PUT THE POWER ON. ON DETECTING POWER UP THE PROGRAM PROCEEDS TO CHECK THAT THE DATA WRITTEN ON CYLINDERS 0 TO 15 WAS INTACT. IF A WRITE CHECKS ERROR OCCURS (POSSIBLY MEANING THAT SOME OF THE DATA WAS DESTROYED DURING THE LOSS OF POWER) IT IS REPORTED AS FOLLOWING:

ERROR, ON POWER-UP RKDA=XXXX

XXXX IS THE CONTENTS OF RKDA AT THE TIME OF ERROR.

THE PROGRAM DOES THE ABOVE POWER FAIL TEST ON ALL DRIVES THAT ARE PRESENT, ONE AFTER THE OTHER IN A ROUND BOBBIN FASHION. EXIT IS THROUGH HALT.

9.9 SECTION SPECIAL

FOR THE BELOW EXAMPLES THE FOLLOWING FORMAT WILL BE USED.
THE ACTUAL TYPEOUT ;COMMENTS ON WHAT
AND RESPONSE ;OCCURRED OR WHAT TO DO
*NOTES IF NECESSARY FOR CLARITY

ERROR EXAMPLE 1

FORMATTER-SURFACE VERIFIER 3

```
                        RK05 CONTROL PANEL TEST          4

                        TYPE=1                           ;TYPE 1 SELECTION.
                        DRIVE NUMBERS ON SYTEM 1=0.      ;DRIVE #0 SELECTED

                        IS THERE A SECOND SYSTEM?N       ;NO SECOND SYSTEM
                        MOUNT PACK ON DRIVE #0           ;CONTINUE PRESSED BUT
                        MAKE PACK WRITE ENABLE           ;WRITE PROTECT ON
                        PRESS CONTINUE WHEN DRIVE RDY    ;CLEAR WRITE PROTECT SWITCH
                        DRIVE WRITE PROTECTED.           ;NOW RUNS TO FINISH
                        DRIVE WRITE PROTECTED            ;THIS DOES NOT EFFECT
                        DONE!                            ;OUTCOME OF TEST
                                RK11 UTILITY PACKAGE

                                NAME                TYPE
                        INDEX                        0
                        COMPATIBILITY PACKAGE        1
                        OSCILLATING SEEK PACKAGE     2

                ERROR EXAMPLE 2

                                RK11 UTILITY PACKAGE

                                NAME                        TYPE
                        INDEX                                0
                        COMPATIBILITY PACKAGE                1
                        OSCILLATING SEEK PACKAGE             2
                        FORMATTER-SURFACE VERIFIER           3
                        RK05 CONTROL PANEL TEST              4
                        RK05 CONTROL PANEL TEST #2           5
                        HEAD ALIGNMENT ROUTINE               6
                        POWER FAILURE (WRITE) TEST           7

                        TYPE=1
                        DRIVE NUMBERS ON SYTEM 1=0.

                        IS THERE A SECOND SYSTEM?N
                        MOUNT PACK ON DRIVE #0
                        MAKE PACK WRITE ENABLE
                        PRESS CONTINUE WHEN DRIVE
                        DRIVE NOT READY
                        DRIVE NOT READY                  ;CONTINUE PRESSED BUT
                        DRIVE NOT READY                  ;DRIVE NOT READY. IF UP
                        DRIVE NOT READY                  ;TO SPEED ETC. AND MESSAGE
                        DRIVE NOT READY                  ;OCCURRING - STATIC SHOULD BE
                        DRIVE NOT READ                   ;RUN IF NOT LOADED OR NOT
                        DRIVE NOT READY                  ;READY MAKING DRIVE READY
                        DRIVE NOT READY                  ;WILL STOP THE MESSAGE
                        DRIVE NOT READY                  ;IT DOES NOT EFFECT THE
                        DRIVE NOT READY
                        DRIVE NOT READY                  ;THE OUTCOME OF COMPATABILITY
                        DRIVE NOT READY
                        DRIVE NOT READY
                        DRIVE NOT READY
                        DRIVE NOT READY
                        DONE!
                                Rk11 UTILITY PACkAGE
```

```
                                        NAME                    TYPE
                                INDEX                           0
                                COMPATIBILITY PACKAGE           1
                                OSCILLATING SEEK PACKAGE        2

                        ERROR EXAMPLE 3

                                RK11 UTILITY PACKAGE

                                        NAME                    TYPE
                                INDEX                           0
                                COMPATIBILITY PACKAGE           1
                                OSCILLATING SEEK PACKAGE        2
                                FORMATTER-SURFACE VERIFIER      3
                                RKOS CONTROL PANEL TEST         4
                                RKOS CONTROL PANEL TEST #2      5
                                HEAD ALIGNMENT ROUTINE          6
                                POWER FAILURE (WRITE) TEST      7

                                TYPE=1                          ;DRIVE RESET TIMED OUT
                                DRIVE NUMBERS ON SYTEM 1=0,1,4,7.

                                IS THERE A SECOND SYSTEM?Y
                                DRIVE # =2
                                MOUNT PACK ON DRIVE #0
                                MAKE PACK WRITE ENABLE          ;THIS MESSAGE IF CONTINUOUS
                                PRESS CONTINUE WHEN DRIVE RDY   ;INDICATED A DRIVE PROBLEM
                                MOUNT PACK ON DRIVE #1          ;THERE IS NO RECOVERY
                                MAKE PACK WRITE ENABLE          ;AND IF CONTINUOUS, A
                                PRESS CONTINUE WHEN DRIVE RDY   ;LOAD START ADDRESS 200
                                DRIVE RESET TIMED OUT           ;IS NECESSARY, DIAGNOSTIC
                                DRIVE RESET TIMED OUT           ;SHOULD BE RUN AGAINST THE
                                DRIVE RESET TIMED OUT           ;FAILING DRIVE.
                                DRIVE RESET TIMED OUT
                                DRIVE RESET TIMED OUT

                                *NOTE A SLOW DRIVE OR FAST PROCESSOR AND MEMORY
                                MAY CAUSE THE MESSAGE TO APPEAR A FEW TIMES AND
                                THEN CONTINUE THIS IS Ok AND WILL NOT EFFECT THE
                                OUTCOME OF THE TEST.

                        ERROR EXAMPLE 4

                                OSCILLATING SEEK PACKAGE        2
                                FORMATTER-SURFACE VERIFIER      3
                                RKOS CONTROL PANEL TEST         4

                                TYPE=1
                                DRIVE NUMBERS ON SYTEM 1=0.

                                IS THERE A SECOND SYSTEM?N      ;SAME AS ABOVE BUT FUNCTION
                                MOUNT PACK ON DRIVE #0          ;WAS A CONTROL RESET
                                MAKE PACK WRITE ENABLE
                                PRESS CONTINUE WHEN DRIVE RDY   ;ALL COMMENTS ARE THE SAME
                                CONTROL RESET TIMED OUT         ;AS EXAMPLE 3
```

```
                                        CONTROL RESET TIMED OUT
                                        CONTROL RESET TIMED OUT
                                        CONTROL RESET TIMED OUT

                                        *A SINGULAR OCCURANCE AS ABOVE IS NOT A PROBLEM
                                        AND WILL NOT EFFECT COMPATABILITY

                                    ERROR EXAMPLE 5

                                                THE BELOW ERRORS DO, ALWAYS, EFFECT COMPATABILITY.
                                                IN THE FIRST TYPE THE DRIVE IS DOWN
                                                INDICATING THAT (5) FIVE HARD OR SOFT
                                                ERRORS OCCURRED.  THE TEST WILL CONTINUE
                                                AGAINST THE OTHER DRIVES BUT THERE IS A
                                                PROBLEM IN THIS DRIVE AND IT SHOULD BE
                                                CONSIDERED NON EXISTENT AS FAR AS COMPATABILITY
                                                GOES.  THAT IS TO SAY IT IS NOT TESTED, THEREFORE
                                                NOT NECESSARILY COMPATABLE OR INCOMPATABLE.

                                                RK11 UTILITY PACKAGE

                                                NAME                    TYPE
                                        INDEX                            0
                                        COMPATIBILITY PACKAGE            1
                                        OSCILLATING SEEK PACKAGE         2
                                        FORMATTER-SURFACE VERIFIER       3
                                        RK05 CONTROL PANEL TEST          4
                                        RK05 CONTROL PANEL TEST #2       5
                                        HEAD ALIGNMENT ROUTINE           6
                                        POWER FAILURE (WRITE) TEST       7

                                        TYPE=1
                                        DRIVE NUMBERS ON SYTEM 1=0.

                                        IS THERE A SECOND SYSTEM?N
                                        MOUNT PACK ON DRIVE #0
                                        MAKE PACK WRITE ENABLE
                                        PRESS CONTINUE WHEN DRIVE RDY
                                        5 ERRORS OCCURRED DRIVE DECLARED DOWN!! NOT TESTED !
                                        DONE!
                                                RK11 UTILITY PACKAGE

                                                NAME                    TYPE
                                        INDEX                            0
                                        COMPATIBILITY PACKAGE            1

                                        *IN THE ABOVE CASE THE MESSAGE "3 SEEK INCOMPLETE
                                        ERRORS OCCURRED DRIVE DECLARED DOWN!! NOT TESTED!"
                                        MAY OCCUR IT IS THE SAME ERROR AS DESCRIBED ABOVE
                                        EXCEPT THAT IT IS CAUSED BY 3 SEEK ERRORS OCCURRING
                                        ON ONE DRIVE.

                                    ERROR EXAMPLE 6

                                                RK11 UTILITY PACKAGE
```

```
                        NAME                    TYPE
INDEX                                            0
COMPATABILITY PACKAGE                            1
OSCILLATING SEEK PACKAGE                         2
FORMATTER-SURFACE VERIFIER                       3
RK05 CONTROL PANEL TEST                          4
RK05 CONTROL PANEL TEST ROUTINE                  5
HEAD ALIGNMENT ROUTINE                           6
POWER FAILURE (WRITE) TEST                       7

TYPE=1
DRIVE NUMBERS ON SYSTEM 1=0.

IS THERE A SECOND SYSTEM?Y
DRIVE # =1
MOUNT PACK ON DRIVE #0
MAKE PACK WRITE ENABLE
PRESS CONTINUE WHEN DRIVE RDY
LOAD AND START ADDRESS 210 ON SYSTEM #2
AND TYPE THE BELOW WHEN ASKED FOR IT.
WORD                1=101000
WORD=000177

MOUNT PACK ON DRIVE #0
MAKE PACK WRITE ENABLE
PRESS CONTINUE WHEN DRIVE RDY
ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
  ADDR=002764  EXPCTD=077400  RECVD=177000
  ADDR=002764  EXPCTD=077400  RECVD=077603
  ADDR=002764  EXPCTD=077400  RECVD=037600
  ADDR=002764  EXPCTD=077400  RECVD=037600
  ADDR=002764  EXPCTD=077400  RECVD=037600
ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
  ADDR=007624  EXPCTD=077400  RECVD=177000
ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
  ADDR=007633  EXPCTD=077400  RECVD=177000
  ADDR=007633  EXPCTD=077400  RECVD=177000
DONE!
        RK11 UTILITY PACKAGE

          NAME                    TYPE
INDEX                              0
COMPATABILITY PACKAGE              1
OSCILLATING SEEK PACKAGE           2
FORMATTER-SURFACE VERIFIER         3
RK05 CONTROL PANEL TEST            4
RK05 CONTROL PANEL TEST #2         5
HEAD ALIGNMENT ROUTINE             6
POWER FAILURE (WRITE) TEST         7

TYPE=
THE ABOVE ERROR MESSAGE SHOWS A COMPATABILITY
PROBLEM.  ALL ERRORS OCCURRED ON HEAD ONE OF
DRIVE 0 TRYING TO READ INFORMATION WRITTEN BY
DRIVE 1.
```

```
                                        ERROR EXAMPLE 7

                                           MOUNT PACK ON DRIVE #0
                                           MAKKE PACK WRITE ENABLE
                                           PRESS CONTINUE WHEN DRIVE RDY
                                           ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
                                             ADDR=000367  EXPCTD=077400  RECVD=077600
                                             ADDR=000367  EXPCTD=077400  RECVD=037600
                                             ADDR=000367  EXPCTD=077400  RECVD=037600
                                             ADDR=000367  EXPCTD=077400  RECVD=037600
                                             ADDR=000367  EXPCTD=077400  RECVD=037600
                                           ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
                                             ADDR=002564  EXPCTD=077400  RECVD=077600
                                             ADDR=002564  EXPCTD=077400  RECVD=037600
                                             ADDR=002564  EXPCTD=077400  RECVD=037600
                                             ADDR=002564  EXPCTD=077400  RECVD=037600
                                             ADDR=002564  EXPCTD=077400  RECVD=037600
                                           ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
                                             ADDR=002764  EXPCTD=077400  RECVD=077600
                                             ADDR=002764  EXPCTD=077400  RECVD=037600
                                             ADDR=002764  EXPCTD=077400  RECVD=037600
                                             ADDR=002764  EXPCTD=077400  RECVD=037600
                                             ADDR=002764  EXPCTD=077400  RECVD=037600
                                           ERROR! DATA WRITTEN BY DRIVE 1 CANNOT BE READ.
                                             ADDR=002767  EXPCTD=077400  RECVD=177000
                                           5 ERRORS OCCURRED DRIVE DECLARED DOWN!! NOT TESTED!
                                           DONE!

                                           IN THE ABOVE EXAMPLE THE PROBLEM IS EXTREME.
                                           THE DRIVE WAS DECLARED DOWN DO TO CHECKSUM
                                           ERRORS.  (TO SEE HOW THIS WAS DETERMINED SEE
                                           PARAGRAPH 9.7).  NOTICE ALSO THE PROBLEM DID NOT
                                           START APPEARING UNTIL CYLINDER 7, AND WAS NOT
                                           FATAL UNTIL CYLINDER 57. AGAIN HEAD #1 WAS A
                                           COMMON FACTOR.
                                                        **********

9.10    COMPATIBILITY ERROR RECOVERY

        ALTHOUGH A UTILITY PACKAGE IS NOT A TRUE DIAGNOSTIC
        IT IS OF BENEFIT TO THE USER TO AT TIMES, BE ABLE
        TO MODIFY THE PROGRAM TO RECIEVE MORE INFORMATION OR
        CONTROL PARAMETERS

        1.  THERE ARE TWO STRATEGICALY PLACED NO-OPS,
            WHICH IF CHANGED TO HALTS, MAY BE OF HELP TO
            THE USER.  ONE IS IN THE 'EXECUTE' ROUTINE WHICH
            ALLOWS THE USER TO EXAMINE THE DISK ADDRESS,
            BUS ADDRESS, WORD COUNT AND CONTROL REGISTERS IN
            TEMPORARY LOCATIONS JUST PRIOR TO LOADING AND
            EXECUTION.  THE SECOND IS IN THE 'ERRCHK' ROUTINE
            WHICH ALLOW THE USER TO EXAMINE THE RKER REGISTER
            BEFORE THE PROGRAM CORRECTS ANY ERRORS WHICH
            WHICH MAY HAVE OCCURRED.

        2.  IF PLAGED BY CHECKSUM ERRORS AND THE USER WISHES
```

```
1046                                        MORE ERROR MAPING THEN HE MAY MODIFY THE
1047                                        MASK WORD AT LOCATION 'ERRCHK+2' TO ONLY RECOGNIZE
1048                                        HARD ERRORS.
1049
1050                                    3.  TO INCREASE OR DECREASE THE NUMBER OF RETRYS ALLOWED
1051                                        BEFORE A DRIVE IS DECLARED DOWN, GO TO THE
1052                                        'MOUNT' ROUTINE, MODIFY THE SETUP OF LOCATIONS
1053                                        'ECNT' AND 'CNTSIN' AND YOU HAVE IT!
1054
1055                                    4.  IF THE USER DECIDES, SAY BECAUSE OF A
1056                                        LARGE NUMBER OF FAILURES, TO ALTER THE NUMBER
1057                                        OF PRINTOUTS PER SECTOR ON FAILURES (THE TYPE IN
1058                                        ERROR EXAMPLE 6 AND 7) HE MAY MODIFY THE SETUP
1059                                        OF 'CHKCNT' IN THE 'RDCHK' ROUTINE.
1060
1061
1062                                    A FINAL LOOK; THE FOLLOWING SECTION SHOWS ALL PACKAGES
1063                                    CALLED IN SEQUENCE, NONE WITH ERRORS.
1064                                            RK11 UTILITY PACKAGE
1065
1066
1067                                                NAME                    TYPE
1068                                    INDEX                               0
1069                                    COMPATABILITY PACKAGE               1
1070                                    OSCILLATING SEEK PACKAGE            2
1071                                    FORMATTER-SURFACE VERIFIER          3
1072                                    RK05 CONTROL PANEL TEST             4
1073                                    RK05 CONTROL PANEL TEST #2          5
1074                                    HEAD ALIGNMENT ROUTINE              6
1075                                    POWER FAILURE (WRITE) TEST          7
1076
1077                                    TYPE=0
1078
1079                                            RK11 UTILITY PACKAGE
1080
1081                                                NAME                    TYPE
1082                                    INDEX                               0
1083                                    COMPATABILITY PACKAGE               1
1084                                    OSCILLATING SEEK PACKAGE
1085                                    FORMATTER-SURFACE VERIFIER
1086                                    RK05 CONTROL PANEL TEST             4
1087                                    RK05 CONTROL PANEL TEST #2          5
1088                                    HEAD ALIGNMENT ROUTINE              6
1089                                    POWER FAILURE (WRITE) TEST          7
1090
1091                                    TYPE=1
1092                                    DRIVE NUMBERS ON SYSTEM 1=0,1,3.
1093
1094                                    IS THERE A SECOND SYSTEM?N
1095                                    MOUNT PACK ON DRIVE #0
1096                                    MAKE PACK WRITE ENABLE
1097                                    PRESS CONTINUE WHEN DRIVE RDY
1098                                    MOUNT PACK ON DRIVE #1
1099                                    MAKE PACK WRITE ENABLE
1100                                    PRESS CONTINUE WHEN DRIVE RDY
1101                                    MOUNT PACK ON DRIVE #3
```

```
1102                                    MAKE PACK WRITE ENABLE
1103                                    PRESS CONTINUE WHEN DRIVE RDY
1104                                    MOUNT PACK ON DRIVE #0
1105                                    MAKE PACK WRITE ENABLE
1106                                    PRESS CONTINUE WHEN DRIVE RDY
1107                                    MOUNT PACK ON DRIVE #1
1108                                    MAKE PACK WRITE ENABLE
1109                                    PRESS CONTINUE WHEN DRIVE RDY
1110                                    MOUNT PACK ON DRIVE #3
1111                                    MAKE PACK WRITE ENABLE
1112                                    PRESS CONTINUE WHEN DRIVE RDY
1113                                    DONE!
1114
1115                                            RK11 UTILITY PACKAGE
1116
1117                                              NAME                    TYPE
1118                                    INDEX                              0
1119                                    COMPATABILITY PACKAGE              1
1120                                    OSCILLATING SEEK PACKAGE           2
1121                                    FORMATTER-SURFACE VERIFIER         3
1122                                    RK05 CONTROL PANEL TEST            4
1123                                    RK05 CONTROL PANEL TEST #2         5
1124                                    HEAD ALIGNMENT ROUTINE             6
1125                                    POWER FAILURE (WRITE) TEST         7
1126
1127                                    TYPE=2
1128                                    OSCILLATING SEEK PACKAGE, WHICH DRIVE?0
1129                                    TOGGLE THE "FIRST CYLINDER ADDRESS" (OUTER LIMIT)
1130                                    INTO THE LOW BYTE (BIT0-7) OF THE SWITCH REGISTER AND THE "LAST
1131                                    CYLINDER ADDRESS" (INNER LIMIT) INTO THE HIGH
1132                                    BYTE (BIT8-15), THEN PRESS CONTINUE.
1133
1134                                            RK11 UTILITY PACKAGE
1135
1136                                              NAME                    TYPE
1137                                    INDEX                              0
1138                                    COMPATABILITY PACKAGE              1
1139                                    OSCILLATING SEEK PACKAGE           2
1140                                    FORMATTER-SURFACE VERIFIER         3
1141                                    RK05 CONTROL PANEL TEST            4
1142                                    RK05 CONTROL PANEL TEST #2         5
1143                                    HEAD ALIGNMENT ROUTINE             6
1144                                    POWER FAILURE (WRITE) TEST         7
1145
1146                                    TYPE=3
1147                                    FORMATTER-SURFACE VERIFIER, WHICH DRIVE?0
1148
1149                                    PACK GOOD.
1150
1151                                            RK11 UTILITY PACKAGE
1152
1153                                              NAME                    TYPE
1154                                    INDEX                              0
1155                                    COMPATABILITY PACKAGE              1
1156                                    OSCILLATING SEEK PACKAGE           2
1157                                    FORMATTER-SURFACE VERIFIER         3
```

```
                                        RK05 CONTROL PANEL TEST          4
                                        RK05 CONTROL PANEL TEST #2       5
                                        HEAD ALIGNMENT ROUTINE           6
                                        POWER FAILURE (WRITE) TEST       7

                                        TYPE=4
                                        RK05 CONTROL PANEL TEST, WHICH DRIVE?0
                                        MOUNT PACK ON DRIVE #0
                                        PLACE DRIVE IN RUN ;SHOULD SEE THE RUN,
                                        POWER, AND ON CYLINDER LAMPS LIGHT.
                                        MAKE DRIVE WRITE ENABLE PRESS CONTINUE

                                        WRITE PROTECT THE DRIVE THEN PRESS CONTINUE

                                        CLEAR WRITE PROTECT THEN PRESS CONTINUE

                                        CAUTION! TRY TO OPEN THE DOOR, DO NOT FORCE:
                                        DOOR SHOULD NOT OPEN!
                                        PRESS CONTINUE WHEN FINISHED

                                        PUT DRIVE IN LOAD, WAIT FOR LOAD LIGHT
                                        PRESS CONTINUE WHEN FINISHED

                                        OPEN THE DOOR, PUT DRIVE IN RUN
                                        CAUTION! IF RUN LIGHT ON ERROR! DEPRESS
                                        LOAD IMMEDIATELY. CONTINUE WHEN FINISHED

                                        REMOVE THE PACK, CLOSE THE DOOR
                                        PUT DRIVE IN RUN, DRIVE SHOULD NOT
                                        RUN...INTERLOCKS HAVE BEEN CHECKED
                                        DONE!

                                                RK11 UTILITY PACKAGE

                                .       %
                                        .ENABLE ABS,AMA
                                .TITLE  MAINDEC-11-DZRKI-C
                                ;*COPYRIGHT (C) 1974
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY BOB COLLINS
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT


                                ;*REVISED BY JIM KAPADIA
                                ;*REVISED BY TOM SAWYER FEB 27, 1976

                                .SBTTL  TRAP CATCHER

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
```

```
                                        ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                        ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                000174                          .=174
000174  000000                         DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                         SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                        .SBTTL  STARTING ADDRESS(ES)
000200  000137  001434                          JMP     @#STARTR        ;;JUMP TO STARTING ADDRESS OF PROGRAM


                000210                          .=210
000210  112737  000377  001312                  MOVB    #377,@#MODE
000216  000137  001440                          JMP     @#START
                                        .SBTTL  BASIC DEFINITIONS

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

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

```
1270            100000                  SW15=   100000
1271            040000                  SW14=   40000
1272            020000                  SW13=   20000
1273            010000                  SW12=   10000
1274            004000                  SW11=   4000
1275            002000                  SW10=   2000
1276            001000                  SW09=   1000
1277            000400                  SW08=   400
1278            000200                  SW07=   200
1279            000100                  SW06=   100
1280            000040                  SW05=   40
1281            000020                  SW04=   20
1282            000010                  SW03=   10
1283            000004                  SW02=   4
1284            000002                  SW01=   2
1285            000001                  SW00=   1
1286                                    .EQUIV  SW09,SW9
1287                                    .EQUIV  SW08,SW8
1288                                    .EQUIV  SW07,SW7
1289                                    .EQUIV  SW06,SW6
1290                                    .EQUIV  SW05,SW5
1291                                    .EQUIV  SW04,SW4
1292                                    .EQUIV  SW03,SW3
1293                                    .EQUIV  SW02,SW2
1294                                    .EQUIV  SW01,SW1
1295                                    .EQUIV  SW00,SW0
1296
1297                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1298         100000                     BIT15=  100000
1299         040000                     BIT14=  40000
1300         020000                     BIT13=  20000
1301         010000                     BIT12=  10000
1302         004000                     BIT11=  4000
1303         002000                     BIT10=  2000
1304         001000                     BIT09=  1000
1305         000400                     BIT08=  400
1306         000200                     BIT07=  200
1307         000100                     BIT06=  100
1308         000040                     BIT05=  40
1309         000020                     BIT04=  20
1310         000010                     BIT03=  10
1311         000004                     BIT02=  4
1312         000002                     BIT01=  2
1313         000001                     BIT00=  1
1314                                    .EQUIV  BIT09,BIT9
1315                                    .EQUIV  BIT08,BIT8
1316                                    .EQUIV  BIT07,BIT7
1317                                    .EQUIV  BIT06,BIT6
1318                                    .EQUIV  BIT05,BIT5
1319                                    .EQUIV  BIT04,BIT4
1320                                    .EQUIV  BIT03,BIT3
1321                                    .EQUIV  BIT02,BIT2
1322                                    .EQUIV  BIT01,BIT1
1323                                    .EQUIV  BIT00,BIT0
1324
1325                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
```

```
1326    000004          ERRVEC= 4           ;;TIME OUT AND OTHER ERRORS
1327    000010          RESVEC= 10          ;;RESERVED AND ILLEGAL INSTRUCTIONS
1328    000014          TBITVEC=14          ;;"T" BIT
1329    000014          TRTVEC= 14          ;;TRACE TRAP
1330    000014          BPTVEC= 14          ;;BREAKPOINT TRAP (BPT)
1331    000020          IOTVEC= 20          ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1332    000024          PWRVEC= 24          ;;POWER FAIL
1333    000030          EMTVEC= 30          ;;EMULATOR TRAP (EMT) **ERROR**
1334    000034          TRAPVEC=34          ;;"TRAP" TRAP
1335    000060          TKVEC=  60          ;;TTY KEYBOARD VECTOR
1336    000064          TPVEC=  64          ;;TTY PRINTER VECTOR
1337    000240          PIRQVEC=240         ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
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  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001161  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001162  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;**********************************************************************
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


001164                          $ERRTB:
001164  000000                  DRACTV: .WORD   0               ;ACTIVE DRIVE WORD

001166  000010                  LOGA:   .BLKW   10              ;TABLE OF ACTIVE DRIVE WORDS

001206  152525                  DRVD:   .WORD   152525          ;TABLE OF PATTERN = TO DRIVE #'S
001210  077777                          .WORD   077777
001212  000000                          .WORD   000000
001214  012345                          .WORD   012345
001216  125252                          .WORD   125252
001220  000001                          .WORD   000001
001222  177777                          .WORD   177777
001224  154320                          .WORD   154320

001236  000010                  ROTBL:  .BLKW   10              ;TABLE OF READ ADDRESS
001246  000004                  PASTBL: .BLKW   4               ;TABLE OF PARAMETERS FOR SYSTEM #2

001256      377                 MSKTBL: .BYTE   377             ;TABLE OF CYLINDER BASE FOR AUTO MODE
001257      177                         .BYTE   177
001260      077                         .BYTE   077
001261      037                         .BYTE   037
001262      017                         .BYTE   017
001263      007                         .BYTE   007
001264      003                         .BYTE   003
001265      001                         .BYTE   001

001266      000                 BASE:   .BYTE   0               ;CYL 0  BASE CYLINDER ADDRESS
001267      050                         .BYTE   50              ;CYL 40 BASE CYLINDER ADDRESS
001270      120                         .BYTE   120             ;CYL 80 BASE CYLINDER ADDRESS
001271      170                         .BYTE   170             ;CYL 120 BASE CYLINDER ADDRESS
001272      240                         .BYTE   240             ;CYL 160 BASE CYLINDER ADDRESS
001273      303                         .BYTE   303             ;CYL 195 BASE CYLINDER ADDRESS

001274  000011                  CYLTBL: .BLKB   11              ;TABLE OF SELECTED BASES

001305      000                 SECTBL: .BYTE   0               ;SECTOR 0
001306      004                         .BYTE   4               ;SECTOR 4
001307      007                         .BYTE   7               ;SECTOR 7
001310      013                         .BYTE   13              ;SECTOR 12

001311      000                 DRCNT1: .BYTE   0               ;COUNT OF NUMBER OF DRIVES ON SYS. 1
001312      000                 MODE:   .BYTE   0               ;IF -1 START 210 SELECTED
```

```
                                PRONUM: .BYTE   0               ;IF 0 1 PROCESSOR SELECTED
                                DRIVE:  .BYTE   0               ;DRIVE # UNDER TEST (MAN+AUTO MODE)
                                CYLBAS: .BYTE   0               ;BASE SELECTED (MANUAL MODE)
                                COMND:  .BYTE   0               ;IF 0 WRITE COMMAND
                                WRTNBY: .BYTE   0               ;DRIVE WHICH DID WRITE (READ OPERATION)
                                HDRFLG: .BYTE   0               ;FLAG FOR ONE HEADER PRINTOUT
                                ECNT:   .BYTE   0               ;ERROR COUNTER
                                CNTSIN: .BYTE   0               ;SEEK INCOM. COUNTER
                                TIMR2:  .BYTE   0               ;SECOND PASS TIMER
                                IDEX:   .BYTE   0               ;CURRENT INDEX #
                                STFLG:  .BYTE   0
                                OSPFLG: .BYTE   0

                        001330          .EVEN

001330  000000          KYTEMP: .WORD   0               ;TEMP. KEYBOARD BUFFER
001332  000000          CONTRL: .WORD   0               ;TEMP. CONTROL+STATUS WORD
001334  000000          DSKADR: .WORD   0               ;TEMP. DISK ADDRESS WORD
001336  000000          BUSADR: .WORD   0               ;TEMP. BUS ADDRESS WORD
001340  000000          WRDCNT: .WORD   0               ;TEMP. WORD COUNT
001342  172000          CYLCNT: .WORD   -6000           ;WORD COUNT OF 1 CYLINDER
001344  177400          SECCNT: .WORD   -400            ;WORD COUNT OF 1 SECTOR
001346  000000          TIMR:   .WORD   0               ;TIMER FOR OPERATIONS
001350  000000          CHKCNT: .WORD   0               ;NUMBER OF ERROR PRINTOUTS
001352  000000          DSKTMP: .WORD   0               ;SAVE OF CURRENT DISK #
001354  004003          WRITCS: .WORD   4003            ;IBA+WRITE+GO
001356  000005          READCS: .WORD   5               ;READ+GO
001360  000000          ERRFLG: .WORD   0               ;ERROR FLAG INHIBIT ADDRESS CHANGE
001362  000000          PATTRN: .WORD   0               ;DATA PATTERN
001364  177400          RKDS:   .WORD   177400
001366  177402          RKER:   .WORD   177402
001370  177404          RKCS:   .WORD   177404
001372  177406          RKWC:   .WORD   177406
001374  177410          RKBA:   .WORD   177410
001376  177412          RKDA:   .WORD   177412
001400  000000          SENDAD: .WORD   0
001402  000000          SEEKI:  .WORD   0
001404  000000          SEEKC:  .WORD   0

                        105212  LFLF=   105212
001406  013620          BA:     BUFF
001410  000000          DA:     .WORD   0
001412  000000          WC:     .WORD   0
001414  013622          RBA:    RBUFF
001416  000000          RWC:    .WORD   0
001420  000000          EXTR:   .WORD   0
001422  000000          ERRWF:  .WORD   0
001424  000000          ERRRF:  .WORD   0
001426  000000          ERRRFC: .WORD   0
001430  000000          ERRWCH: .WORD   0
001432  000000          ERRWCS: .WORD   0

                                ;BIT DEFINITIONS

                        010000  DPL=BIT12
                        000100  RWS=BIT6
```

```
                000040                  WPS=BIT5
                001000                  SIN=BIT9

001434  105037  001312         STARTR: CLRB    @#MODE
001440                         START:
                               .SBTTL  INITIALIZE THE COMMON TAGS
                               ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001440  012706  001100                 MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001444  005026                 1$:     CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001446  022706  001140                 CMP     #SWR,R6 ;;DONE?
001452  001374                         BNE     .-6             ;;LOOP BACK IF NO
001454  012706  001100                 MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                               ;;INITIALIZE A FEW VECTORS
001460  012737  023220  000034         MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001466  012737  000340  000036         MOV     #340,@#TRAPVEC+2;LEVEL 7
001474  012737  023274  000024         MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001502  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;LEVEL 7
                               ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                               ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001510  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001514  012737  001550  000004         MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001522  012737  177570  001140         MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001530  012737  177570  001142         MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001536  022777  177777  177374         CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001544  001012                         BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                               ;;AND  THE HARDWARE SWR IS NOT = -1
001546  000403                         BR      65$             ;;BRANCH IF NO TIMEOUT
001550  012716  001556         64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001554  000002                         RTI
001556  012737  000176  001140 65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001564  012737  000174  001142         MOV     #DISPREG,DISPLAY
001572  012637  000004         66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001576  105737  001312                 TSTB    @#MODE
001602  100002                         BPL     1$
001604  000137  003412                 JMP     @#SECOND
001610  105737  001325         1$:     TSTB    STFLG                   ;PRINT ONLY THE FIRST TIME
001614  001402                         BEQ     10$
001616  000137  002476                 JMP     TABLTY
001622  105237  001325         10$:    INCB    STFLG
001626  105037  001326         STRT1:  CLRB    OSPFLG
                               .SBTTL  TYPE PROGRAM NAME
                               ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
001632  005227  177777                 INC     #-1             ;;FIRST TIME?
001636  001031                         BNE     64$             ;;BRANCH IF NO
001640  104401  001646                 TYPE    ,65$            ;;TYPE ASCIZ STRING
001644  000426                         BR      64$             ;;GET OVER THE ASCIZ
                               ;;65$:  .ASCIZ  <CRLF><15><12>/MAINDEC-11-DZRKI-C RK11 UTILITY PACKAGE/<CRLF>
001722                         64$:
001722  104401  001730                 TYPE    ,69$            ;;TYPE ASCIZ STRING
001726  000423                         BR      68$             ;;GET OVER THE ASCIZ
                               ;;69$:  .ASCIZ  <15><12><15><12>/        NAME                    TYPE
001776                         68$:
001776  104401  002004                 TYPE    ,71$            ;;TYPE ASCIZ STRING
002002  000421                         BR      70$             ;;GET OVER THE ASCIZ
                               ;;71$:  .ASCIZ  <15><12>/INDEX                          0
```

```
1547  002046
1548  002046  104401  002054
1549  002052  000421
1550
1551  002116
1552  002116  104401  002124
1553  002122  000421
1554
1555  002166
1556  002166  104401  002174
1557  002172  000421
1558
1559  002236
1560  002236  104401  002244
1561  002242  000421
1562
1563  002306
1564  002306  104401  002314
1565  002312  000421
1566
1567  002356
1568  002356  104401  002364
1569  002362  000421
1570
1571  002426
1572  002426  104401  002434
1573  002432  000421
1574
1575  002476
1576  002476
1577  002476  104401  002504
1578  002502  000405
1579
1580  002516
1581  002516  104405
1582  002520  011600
1583  002522  104403
1584  002524     001
1585  002525     000
1586  002526  162700  000060
1587  002532  100406
1588  002534  022700  000010
1589  002540  003403
1590  002542  006100
1591  002544  000170  002562
1592  002550
1593  002550  104401  002556
1594  002554  000401
1595
1596  002560
1597  002560  000746
1598  002562  001626
1599  002564  002602
1600  002566  010330
1601  002570  011712
1602  002572  013712
```

```
70$:
        TYPE    ,73$            ;;TYPE ASCIZ STRING
        BR      72$             ;;GET OVER THE ASCIZ
;;73$:  .ASCIZ  <15><12>/COMPATIBILITY PACKAGE          1/
72$:
        TYPE    ,75$            ;;TYPE ASCIZ STRING
        BR      74$             ;;GET OVER THE ASCIZ
;;75$:  .ASCIZ  <15><12>/OSCILLATING SEEK PACKAGE       2/
74$:
        TYPE    ,77$            ;;TYPE ASCIZ STRING
        BR      76$             ;;GET OVER THE ASCIZ
;;77$:  .ASCIZ  <15><12>/FORMATTER-SURFACE VERIFIER     3/
76$:
        TYPE    ,79$            ;;TYPE ASCIZ STRING
        BR      78$             ;;GET OVER THE ASCIZ
;;79$:  .ASCIZ  <15><12>/RK05 CONTROL PANEL TEST        4/
78$:
        TYPE    ,81$            ;;TYPE ASCIZ STRING
        BR      80$             ;;GET OVER THE ASCIZ
;;81$:  .ASCIZ  <15><12>/RK05 CONTROL PANEL TEST #2     5/
80$:
        TYPE    ,83$            ;;TYPE ASCIZ STRING
        BR      82$             ;;GET OVER THE ASCIZ
;;83$:  .ASCIZ  <15><12>/HEAD ALIGNMENT ROUTINE         6/
82$:
        TYPE    ,85$            ;;TYPE ASCIZ STRING
        BR      84$             ;;GET OVER THE ASCIZ
;;85$:  .ASCIZ  <15><12>/POWER FAILURE (WRITE) TEST     7/
84$:
TABLTY:
        TYPE    ,65$            ;;TYPE ASCIZ STRING
        BR      64$             ;;GET OVER THE ASCIZ
;;65$:  .ASCIZ  <15><12  15><12>/TYPE=/
64$:
        RDCHR
        MOV     (SP),R0
        TYPOS
        .BYTE   1
        .BYTE   0
        SUB     #60,R0
        BMI     NG
        CMP     #10,R0
        BLE     NG
        ROL     R0
        JMP     @BEGIN(R0)
NG:
        TYPE    ,65$            ;;TYPE ASCIZ STRING
        BR      64$             ;;GET OVER THE ASCIZ
;;65$:  .ASCIZ  /?/
64$:
        BR      TABLTY
BEGIN:  START
        SECT.3
        SECT.2
        SECT.1
        SECT.0
```

```
                                        SECT.4
                                        SECT.5
                                        SECT.6


                                        .SBTTL  COMPATIBILITY TEST

                                ;ROUTINE TO PICK UP THE DRIVE NUMBER TO BE TESTED
                                ;ON SYSTEM 1.

                                SECT.3: NOP                     ;NO-OP
                                AUTSL2:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/TERMINATE WITH '.<CR>'/
                                64$:
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/DRIVE NUMBERS ON SYSTEM 1=/
                                66$:
                                        RDLIN
                                        MOV     (SP)+,R0        ;PICK UP THE ADDRESS OF THE INPUT BUFFER
                                        MOV     #LOGA,R1        ;GET THE ADDRESS OF THE LOGICAL UNIT TBL.
                                        CLRB    @#DRCNT1        ;CLEAR THE DRIVE COUNTER
                                1$:     CLR     @#KYTEMP        ;CLEAR TEMP
                                        MOVB    (R0)+,@#KYTEMP  ;GET THE FIRST DRIVE #
                                        CMPB    #54,@#KYTEMP    ;IS IT A COMMA THAT WAS TYPED?
                                        BEQ     1$              ;IF YES GO BACK
                                        SUB     #60,@#KYTEMP    ;MAKE ASCII A DRIVE #
                                        BMI     2$              ;IF RESULT NEGATIVE BRANCH
                                        JSR     PC,STORE        ;IF RESULT POSITIVE JUMP
                                        BR      1$              ;AFTER STORING GET NEXT #
                                2$:     CMPB    #56,-(R0)       ;WAS NEGATIVE RESULT A TERMINATOR?
                                        BEQ     3$              ;IF YES BRANCH
                                        JSR     PC,ILEGAL       ;IF NO BAD CHARACTER JUMP
                                3$:     CMP     #DRV0,R1        ;IS THE TABLE FULL
                                        BEQ     SECSYS          ;IF YES BRANCH
                                        MOV     #100000,(R1)+   ;IF NO FILL TABLE WITH DOWN INDICATOR.
                                        BR      3$              ;GO BACK AND CHECK

                                ;ROUTINE TO DETERMINE IF THERE IS A SECOND
                                ;SYSTEM AND IF SO TO GET THE NUMBER OF THE
                                ;DRIVE ON THIS SYSTEM

                                SECSYS:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/IS THERE A SECOND SYSTEM?/
                                64$:
                                        NOP                     ;***
                                        RDCHR                   ;READ A CHARACTER
                                        MOV     (SP)+,@#KYTEMP  ;GET THE RESPONSE
                                        TYPE    ,KYTEMP         ;ECHO
                                        CMP     #131,@#KYTEMP   ;WAS IT A "Y" (FOR YES)?
                                        BEQ     PRO2            ;IF YES BRANCH TO PROCESSOR 2
                                        CMP     #116,@#KYTEMP   ;WAS RESPONSE LEGAL (N FOR NO)?
```

```
1659  003104  001460                          BEQ     PRO1            ;IF LEGAL BRANCH
1660  003106  104401  003114                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1661  003112  000401                          BR      66$             ;;GET OVER THE ASCIZ
1662                                  ;;67$:  .ASCIZ  /?/
1663  003116                          66$:
1664  003116  000734                          BR      SECSYS          ;GO BACK ASK AGAIN
1665  003120  152737  000377  001313  PRO2:   BISB    #377,@#PRONUM   ;SET FLAG TWO PROCESSORS
1666  003126  000240                          NOP                     ;***
1667  003130  104401  003136                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1668  003134  000406                          BR      64$             ;;GET OVER THE ASCIZ
1669                                  ;;65$:  .ASCIZ  <15><12>/DRIVE #"=/
1670  003152                          64$:
1671  003152  104405                          RDCHR                   ;READ A CHARACTER
1672  003154  012637  001330                  MOV     (SP)+,@#KYTEMP  ;PICK UP THE RESPONSE
1673  003160  104401  001330                  TYPE    ,KYTEMP         ;ECHO
1674  003164  162737  000060  001330          SUB     #60,@#KYTEMP    ;MAKE IT A NUMBER
1675  003172  100420                          BMI     BADINP          ;IF NOT A NUMBER, BRANCH
1676  003174  022737  000010  001330          CMP     #10,@#KYTEMP    ;IS IT A LEGAL #?
1677  003202  003414                          BLE     BADINP          ;IF NO BRANCH
1678  003204  000337  001330                  SWAB    @#KYTEMP        ;GET THE DRIVE # TO THE HIGH BYTE
1679  003210  052737  040000  001330          BIS     #BIT14,@#KYTEMP ;SET THE SECOND SYSTEM BIT
1680  003216  113705  001311                  MOVB    @#DRCNT1,R5     ;GET THE DRIVE COUNT
1681  003222  006105                          ROL     R5              ;MAKE IT AN INDEX
1682  003224  013765  001330  001166          MOV     @#KYTEMP,LOCA(R5) ;STORE THE SYSTEM #2 WORD
1683  003232  000407                          BR      GO              ;GO DO THE TEST
1684  003234                          BADINP:
1685  003234  104401  003242                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1686  003240  000401                          BR      64$             ;;GET OVER THE ASCIZ
1687                                  ;;65$:  .ASCIZ  /?/
1688  003244                          64$:
1689  003244  000725                          BR      PRO2            ;GO BACK ASK AGAIN
1690  003246  105037  001313          PRO1:   CLRB    @#PRONUM        ;CLEAR THE FLAG ONE PROCESSOR
1691
1692                                  ;THIS IS THE ACTUAL PROGRAM
1693
1694  003252  012700  001166          GO:     MOV     #LOCA,R0        ;GET THE TABLE ADDRESS TO R0
1695  003256  105037  001324                  CLRB    @#IDEX          ;CLRB THE INDEX
1696  003262  000405                          BR      GO2             ;BRANCH AROUND INCREMENT ROUTINE
1697
1698  003264  062700  000002          GO1:    ADD     #2,R0           ;INDEX THRU THE TABLE
1699  003270  022700  001206                  CMP     #DRV0,R0        ;DONE?
1700  003274  001414                          BEQ     EXIT            ;IF YES GET OUT
1701  003276  005710                  GO2:    TST     (R0)            ;IS THE DRIVE ACTIVE
1702  003300  100001                          BPL     GO3             ;IF YES BRANCH
1703  003302  000770                          BR      GO1             ;NO TRY THE NEXT ONE
1704  003304  011037  001164          GO3:    MOV     (R0),@#DRACTV   ;PICK UP THE ACTIVE DRIVE WORD
1705  003310  004737  004134                  JSR     PC,CYCLE        ;CALL CYCLE (PICK UP DRIVE #, CYL BASE, AND CALL MOUNT)
1706  003314  004737  005044                  JSR     PC,WRLINK       ;CALL WRITE LINK TO LOAD REGISTERS FOR WRITE
1707  003320  004737  005572                  JSR     PC,RDLINK       ;CALL READ LINK TO LOAD REGISTERS FOR READ
1708  003324  000757                          BR      GO1             ;GO GET NEXT DRIVE
1709  003326  012700  001166          EXIT:   MOV     #LOCA,R0
1710  003332  022700  001206          EXTFR2: CMP     #DRV0,R0
1711  003336  001414                          BEQ     EXITX
1712  003340  032710  040000                  BIT     #BIT14,(R0)
1713  003344  001011                          BNE     EXITX
1714  003346  005720                          TST     (R0)+
```

```
1715  003350  100770                          BMI     EXTFR2
1716  003352  014001                          MOV     -(R0),R1
1717  003354  004737  004210                  JSR     PC,D02
1718  003360  004737  005572                  JSR     PC,RDLINK
1719  003364  005720                          TST     (R0)+
1720  003366  000761                          BR      EXTFR2
1721  003370                          EXITX:
1722  003370  104401  003376                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1723  003374  000404                          BR      64$             ;;GET OVER THE ASCIZ
1724                                  ;;65$:  .ASCIZ  <15><12>/DONE!/
1725  003406                          64$:
1726  003406  000137  001440                  JMP     @#START         ;RESTART
1727
1728                                  ;THIS IS PROCESSOR #2 CODE. THIS ROUTINE ASKS FOR AND UNPACKS THE CONTROL
1729                                  ;WORDS FROM THE FIRST PROCESSOR.
1730
1731  003412                          SECOND:
1732  003412  104401  003420                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1733  003416  000415                          BR      64$             ;;GET OVER THE ASCIZ
1734                                  ;;65$:  .ASCIZ  <15><12>/COMPATIBILITY-SYSTEM#2/
1735  003452                          64$:
1736  003452  012704  000200                  MOV     #200,R4         ;SET UP MASK BIT IN R4
1737  003456  012703  000001                  MOV     #1,R3           ;SET UP WORD COUNTER IN R3
1738  003462  012702  001166                  MOV     #LOGA,R2        ;GET THE TABLE ADDRESS
1739  003466                          INSYS2:
1740  003466  104401  003474                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1741  003472  000405                          BR      64$             ;;GET OVER THE ASCIZ
1742                                  ;;65$:  .ASCIZ  <15><12>/WORD  /
1743  003506                          64$:
1744  003506  000240                          NOP                     ;***
1745  003510  010346                          MOV     R3,-(SP)        ;GET READY TO TYPE WORD COUNTER
1746  003512  104403                          TYPOS
1747  003514     006                          .BYTE   6
1748  003515     000                          .BYTE   0
1749  003516  104401  003524                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1750  003522  000401                          BR      66$             ;;GET OVER THE ASCIZ
1751                                  ;;67$:  .ASCIZ  /=/
1752  003526                          66$:
1753  003526  104407                          RDOCT
1754  003530  012600                          MOV     (SP)+,R0        ;PICK UP THE OCTAL WORD
1755  003532  110001                          MOVB    R0,R1           ;GET THE FIRST DRIVE TO R1
1756  003534  000300                          SWAB    R0              ;GET THE SECOND DRIVE TO R0 LOW BYTE
1757  003536  042700  177400                  BIC     #177400,R0      ;CLEAR THE UNUSED BITS
1758  003542  042701  177400                  BIC     #177400,R1      ;CLEAR THE UNUSED BITS
1759  003546  006000                          ROR     R0              ;ROTATE RIGHT R0
1760  003550  103003                          BCC     2$              ;IF CARRY IS CLEAR BRANCH
1761  003552  052700  000200                  BIS     #BIT7,R0        ;SET DOWN BIT IF CARRY SET
1762  003556  000241                          CLC                     ;AND CLEAR THE CARRY BIT
1763  003560  006001                  2$:     ROR     R1              ;NOW DO THE SAME FOR R1
1764  003562  103002                          BCC     3$              ;IF NO ERROR, BRANCH
1765  003564  052701  000200                  BIS     #BIT7,R1        ;IF ERROR SET THE BIT
1766  003570  110162  000001          3$:     MOVB    R1,1(R2)        ;SET DRIVE FIRST IN TABLE
1767  003574  004737  003770                  JSR     PC,MASKER       ;CALL THE MASK CONTROL SUBROUTINE
1768  003600  110062  000001                  MOVB    R0,1(R2)        ;SET DRIVE SECOND IN TABLE (NEXT WORD)
1769  003604  004737  003770                  JSR     PC,MASKER       ;CALL THE MASK CONTROL SUBROUTINE
1770  003610  005203                          INC     R3              ;INCREMENT THE WORD COUNTER
```

```
 1771  003612  000725                          BR      INSYS2          ;GO BACK AND GET NEXT WORD
 1772  003614  050412                  EXITA:  BIS     R4,(R2)         ;FILL THE TABLE (LAST WORD)
 1773  003616  006004                          ROR     R4              ;WITH ALL BITS SET
 1774  003620  103375                          BCC     EXITA           ;GO BACK IF NOT DONE
 1775  003622  042712  174000          EXITB:  BIC     #174000,(R2)    ;CLEAR DOWN AND SECOND SYSTEM BITS AT TABLE
 1776  003626  010200                          MOV     R2,R0           ;GET ADDRESS TO R0 (CURRENT TABLE)
 1777  003630  005722                          TST     (R2)+           ;ADD 2 TO THE POINTER
 1778  003632  022702  001206          FIL.DN: CMP     #DRV0,R2        ;TABLE FULL?
 1779  003636  001403                          BEQ     2$              ;IF YES BRANCH
 1780  003640  012722  100000                  MOV     #BIT15,(R2)+    ;FILL THE TABLE
 1781  003644  000772                          BR      FIL.DN          ;GO BACK TRY AGAIN
 1782  003646  011001                  2$:     MOV     (R0),R1         ;GET THE WORD TO R1
 1783  003650  110102                          MOVB    R1,R2
 1784  003652  004737  005174                  JSR     PC,MASK
 1785  003656  004737  004210                  JSR     PC,D02          ;FORM DRIVE # FOR MOUNT
 1786  003662  004737  005044                  JSR     PC,WRLINK       ;WRITE NEW INFO
 1787  003666  004737  005572                  JSR     PC,RDLINK       ;READ SAMPLE (ALL DRIVES)
 1788  003672  104401  003700                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 1789  003676  000427                          BR      64$             ;;GET OVER THE ASCIZ
 1790                          ;;65$:  .ASCIZ  <15><12>/DONE SYSTEM 2 RESTART SYSTEM 1, TYPE WORD /
 1791  003756                  64$:
 1792  003756  011046                          MOV     (R0),-(SP)      ;GET WORD FOR SYSTEM 1 AND TYPE
 1793  003760  104403                          TYPOS
 1794  003762     006                          .BYTE   6
 1795  003763     001                          .BYTE   1
 1796  003764  000000                  1$:     HALT                    ;HALT (FINISHED SYSTEM #2)
 1797  003766  000776                          BR      1$              ;GO BACK TO HALT
 1798
 1799                          ;THIS ROUTINE SETS UP THE MASK BITS IN THE LOG TABLE FOR
 1800                          ;SYSTEM #2, IF A DRIVE IS DOWN NO BIT IS SET BUT THE MASK
 1801                          ;IS SHIFTED.
 1802
 1803  003770  005712          MASKER: TST     (R2)            ;IS THE DRIVE UP
 1804  003772  100401                  BMI     RETRN4          ;IF NO, BRANCH
 1805  003774  110412                  MOVB    R4,(R2)         ;MOVE THE MASK BIT IN THE TABLE
 1806  003776  006004          RETRN4: ROR     R4              ;ROTATE THE MASK
 1807  004000  103003                  BCC     1$              ;DONE?,IF NO,BRANCH
 1808  004002  012716  003622          MOV     #EXITB,(SP)     ;SET UP FOR RETURN
 1809  004006  000207                  RTS     PC
 1810  004010  032712  040000  1$:     BIT     #BIT14,(R2)     ;IS THIS SYSTEM # 2'S DRIVE?
 1811  004014  001403                  BEQ     2$              ;IF NO,BRANCH
 1812  004016  012716  003614          MOV     #EXITA,(SP)     ;SET UP FOR RETURN
 1813  004022  000207                  RTS     PC
 1814  004024  062702  000002  2$:     ADD     #2,R2           ;INDEX THRU LOGA TABLE
 1815  004030  000207                  RTS     PC              ;RETURN
 1816
 1817                          ;ROUTINE TO BUILD THE ACTIVE LOGICAL UNIT TABLE
 1818
 1819  004032  022737  000010  001330  STORE:  CMP     #10,@#KYTEMP    ;IS INPUT A LEGAL NUMBER?
 1820  004040  000240                  NOP                     ;***
 1821  004042  003417                  BLE     ILEGAL          ;IF NOT BRANCH
 1822  004044  000337  001330          SWAB    @#KYTEMP        ;ALIGN DRIVE # FOR TABLE
 1823  004050  013721  001330          MOV     @#KYTEMP,(R1)+  ;PUT THE WORD IN THE TABLE
 1824  004054  005037  001330          CLR     @#KYTEMP        ;CLEAR THE TEMP WORD
 1825  004060  105237  001311          INCB    @#DRCNT1        ;INCREMENT THE COUNTER
 1826  004064  022701  001206          CMP     #DRV0,R1        ;IS THE TABLE FULL?
```

```
1827  004070  001401                        BEQ     TBLFUL          ;IF YES BRANCH
1828  004072  000207                        RTS     PC              ;IF NO RETURN
1829  004074  012716  003010      TBLFUL:   MOV     #SECSYS,(SP)    ;IF TABLE FULL SET UP FOR SYSTEM #2
1830  004100  000207                        RTS     PC              ;RETURN
1831  004102  012716  002604      ILEGAL:   MOV     #AUTSL2,(SP)    ;IF ILLEGAL RESPONSE, GO BACK
1832  004106  104401  004114              TYPE    ,65$            ;;TYPE ASCIZ STRING
1833  004112  000407                        BR      64$             ;;GET OVER THE ASCIZ
1834                                ;;65$:  .ASCIZ  /ILLEGAL INPUT/
1835  004132                        64$:
1836  004132  000207                        RTS     PC              ;RETURN
1837
1838                                ;THIS ROUTINE GETS A DRIVE # AND BUILDS A TABLE OF
1839                                ;CYLINDER ADDRESS FOR USE BY WRITE. (R0)=ADDRESS OF ACTIVE
1840                                ;WORD IN LOGICAL TABLE. IT ALSO SETS AND CLEARS THE MASK BITS
1841                                ;OF THE TABLE AS OPERATIONS INDICATE
1842
1843  004134  011001      CYCLE:    MOV     (R0),R1         ;GET THE LOGICAL UNIT ACTIVE WORD TO R1
1844  004136  032701  040000        BIT     #BIT14,R1       ;IS IT ON SYSTEM #2
1845  004142  001402                BEQ     CYCL2           ;IF NO BRANCH
1846  004144  000137  006712        JMP     @#SECONE        ;GO TO SYSTEM #2
1847  004150  113703  001324  CYCL2:   MOVB    @#IDEX,R3       ;GET THE INDEX VALUE
1848  004154  116302  001256        MOVB    MSKTBL(R3),R2   ;GET THE MASK TO R2
1849  004160  004737  005174  CYCLE2:  JSR     PC,MASK         ;CALL THE MASK SUBROUTINE
1850  004164  012703  001166  LDFLG:   MOV     #LOGA,R3        ;GET TABLE ADDRESS TO R3
1851  004170  020003          2$:      CMP     R0,R3           ;IS ACTIVE ADDRESS=TO FIRST ADDRESS
1852  004172  001002                BNE     3$              ;IF NO BRANCH
1853  004174  050223                BIS     R2,(R3)+        ;IF YES SET BITS TO SHOW WRITE
1854  004176  000401                BR      4$              ;BRANCH TO SEE IF DONE
1855  004200  040223          3$:      BIC     R2,(R3)+        ;CLEAR BITS TO SHOW OVER-WRITE
1856  004202  022703  001206  4$:      CMP     #DRV0,R3        ;DONE
1857  004206  001370                BNE     2$              ;IF NO GO BACK
1858  004210  000301          D02:     SWAB    R1              ;GET DRIVE # TO LOW BYTE
1859  004212  000240                NOP                     ;***
1860  004214  110102                MOVB    R1,R2           ;GET IT TO R2
1861  004216  042702  000370        BIC     #370,R2         ;CLEAR THE UNUSED BITS
1862  004222  110237  001314        MOVB    R2,@#DRIVE      ;GET THE DRIVE #
1863  004226  006102                ROL     R2              ;SHIFT THE DRIVE # TO
1864  004230  006102                ROL     R2              ;ALIGN IT FOR THE DRIVE ADDR.
1865  004232  006102                ROL     R2              ;KEEP IT MOVING!
1866  004234  006102                ROL     R2              ;A LITTLE MORE!
1867  004236  006102                ROL     R2              ;THERE IT IS
1868  004240  110237  001353        MOVB    R2,@#DSKTMP+1
1869  004244  110237  001335        MOVB    R2,@#DSKADR+1   ;GET IT TO DISK ADDR. TEMP.
1870  004250  004737  004256        JSR     PC,MOUNT        ;CALL MOUNT
1871  004254  000207                RTS     PC              ;RETURN
1872
1873  004256  105237  001324  MOUNT:   INCB    @#IDEX          ;INCREMENT THE INDEX
1874  004262  000240                NOP                     ;***
1875  004264  112737  000005  001321  MOVB    #5,@#ECNT       ;SET ERROR CNTR TO 5
1876  004272  112737  000003  001322  MOVB    #3,@#CNTSIN     ;SET SIN CNTR TO 3
1877  004300  104401  004306        TYPE    ,65$            ;;TYPE ASCIZ STRING
1878  004304  000414                BR      64$             ;;GET OVER THE ASCIZ
1879                                ;;65$:  .ASCIZ  <15><12>/MOUNT PACK ON DRIVE #/
1880  004336                        64$:
1881  004336  013746  001314        MOV     DRIVE,-(SP)     ;;SAVE DRIVE FOR TYPEOUT
1882  004342  104403                TYPOS                   ;;GO TYPE--OCTAL ASCII
```

```
 1883  004344  001                             .BYTE   1               ;;TYPE 1 DIGIT(S)
 1884  004345  000                             .BYTE   0               ;;SUPPRESS LEADING ZEROS
 1885  004346  104401  004354                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 1886  004352  000415                          BR      66$             ;;GET OVER THE ASCIZ
 1887                                  ;;67$:  .ASCIZ  <15><12>/MAKE PACK WRITE ENABLE/
 1888  004406                          66$:
 1889  004406  104401  004414                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 1890  004412  000420                          BR      68$             ;;GET OVER THE ASCIZ
 1891                                  ;;69$:  .ASCIZ  <15><12>/PRESS CONTINUE WHEN DRIVE RDY/
 1892  004454                          68$:
 1893  004454  000000                          HALT
 1894  004456  004737  004464                  JSR     PC,INITIL       ;CALL INITIALIZER
 1895  004462  000207                          RTS     PC              ;RETURN
 1896
 1897                                  ;THIS ROUTINE INITIALIZES A DRIVE AND INSURES THAT IT IS READY AND
 1898                                  ;WRITE ENABLED. IT IS ENTERED FROM MOUNT OR FROM EXECUTE IF OPERATION FAILS
 1899
 1900  004464  010046                  INITIL: MOV     R0,-(SP)        ;SAVE R0
 1901  004466  013700  001334                  MOV     @#DSKADR,R0     ;GET THE DRIVE # TO R0
 1902  004472  042700  001777                  BIC     #1777,R0        ;CLEAR THE UNUSED BITS
 1903  004476  005037  001346          INITI2: CLR     @#TIMR          ;CLEAR THE TIMER
 1904  004502  105037  001323                  CLRB    @#TIMR2
 1905  004506  000240                          NOP                     ;***
 1906  004510  010077  174662                  MOV     R0,@RKDA        ;GET DRIVE # TO 'DA' REGISTER
 1907  004514  012777  000001  174646          MOV     #1,@RKCS        ;ISSUE CONTROL RESET + GO
 1908  004522  004737  005554                  JSR     PC,SMTME
 1909  004526  105777  174636          1$:     TSTB    @RKCS           ;DID CONTROL READY SET
 1910  004532  100423                          BMI     2$              ;IF YES BRANCH
 1911  004534  005237  001346                  INC     @#TIMR          ;IF NO INCREMENT THE TIMER
 1912  004540  001372                          BNE     1$              ;IF TIMER NOT ZERO BRANCH
 1913  004542  104401  004550                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 1914  004546  000415                          BR      64$             ;;GET OVER THE ASCIZ
 1915                                  ;;65$:  .ASCIZ  <15><12>/CONTROL RESET TIME OUT/
 1916  004602                          64$:
 1917  004602  010077  174570          2$:     MOV     R0,@RKDA        ;DRIVE NUMBER TO 'DA' REG.
 1918  004606  105777  174552                  TSTB    @RKDS           ;IS DRIVE READY
 1919  004612  100415                          BMI     3$              ;IF YES BRANCH
 1920  004614  104401  004622                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 1921  004620  000411                          BR      66$             ;;GET OVER THE ASCIZ
 1922                                  ;;67$:  .ASCIZ  <15><12>/DRIVE NOT READY/
 1923  004644                          66$:
 1924  004644  000714                          BR      INITI2          ;GO BACK TRY AGAIN
 1925  004646  032777  000040  174510  3$:     BIT     #BIT5,@RKDS     ;IS DRIVE WRITE LOCKED?
 1926  004654  001420                          BEQ     4$              ;IF NO, BRANCH
 1927  004656  104401  004664                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 1928  004662  000414                          BR      68$             ;;GET OVER THE ASCIZ
 1929                                  ;;69$:  .ASCIZ  <15><12>/DRIVE WRITE PROTECTED/
 1930  004714                          68$:
 1931  004714  000670                          BR      INITI2          ;YES, GO BACK TRY AGAIN
 1932  004716  005037  001346          4$:     CLR     @#TIMR          ;CLEAR THE TIMER
 1933  004722  010077  174450                  MOV     R0,@RKDA        ;GET THE DRIVE # TO 'DA' REGISTER
 1934  004726  012777  000015  174434          MOV     #15,@RKCS       ;ISSUE DRIVE RESET + GO
 1935  004734  004737  005554                  JSR     PC,SMTME
 1936  004740  105777  174424          5$:     TSTB    @RKCS
 1937  004744  100375                          BPL     5$
 1938  004746  032777  000100  174410          BIT     #100,@RKDS      ;READ/WRITE/SEEK READY BIT SET?
```

```
 1939  004754  001031                          BNE     6$              ;IF YES, BRANCH
 1940  004756  005237  001346                  INC     @#TIMR          ;NO, INCREMENT THE TIMER
 1941  004762  001366                          BNE     5$              ;GO BACK AND CHECK IF TIMER NOT 0
 1942  004764  105737  001323                  TSTB    @#TIMR2
 1943  004770  001003                          BNE     7$
 1944  004772  105237  001323                  INCB    @#TIMR2
 1945  004776  000760                          BR      5$
 1946  005000                          7$:
 1947  005000  104401  005006                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 1948  005004  000414                          BR      70$             ;;GET OVER THE ASCIZ
 1949                                  ;;71$:   .ASCIZ  <15><12>/DRIVE RESET TIMED OUT/
 1950  005036                          70$:
 1951  005036  000727                          BR      4$              ;GO BACK, TRY AGAIN
 1952  005040  012600                  6$:     MOV     (SP)+,R0        ;RESTORE R0
 1953  005042  000207                          RTS     PC              ;RETURN TO CALLER
 1954
 1955                                  ;THIS ROUTINE TAKES CARE OF ALL LINKAGES FOR THE
 1956                                  ;EXECUTE ROUTINE.  IT FORMS THE ADDRESS AND SETS UP ALL THE
 1957                                  ;REGISTERS FOR THE WRITE OPERATION
 1958
 1959  005044  105037  001316          WRLINK: CLRB    @#COMND         ;INDICATE WRITE OPERATION
 1960  005050  000240                          NOP                     ;***
 1961  005052  113701  001314                  MOVB    @#DRIVE,R1      ;PICK UP THE DRIVE #
 1962  005056  006101                          ROL     R1              ;MAKE IT A WORD INDEX
 1963  005060  016137  001206  001362  1$:     MOV     DRVO(R1),@#PATTRN ;PICK UP THE DATA PATTERN
 1964  005066  004737  005242                  JSR     PC,CYLADR       ;CALL CYLINDER ADDRESS
 1965  005072  000401                          BR      2$              ;RETURN HERE IF NOT LAST BASE
 1966  005074  000207                          RTS     PC              ;RETURN HERE IF LAST BASE
 1967  005076  012737  001362  001336  2$:     MOV     #PATTRN,@#BUSADR ;GET THE ADDRESS OF THE OUTPUT
 1968  005104  013737  001342  001340          MOV     @#CYLCNT,@#WRDCNT ;GET THE WORD COUNT
 1969  005112  013737  001354  001332          MOV     @#WRITCS,@#CONTRL ;GET THE CONTROL + STATUS WORD
 1970  005120  004737  005342                  JSR     PC,EXECUT       ;CALL EXECUTE
 1971  005124  032737  000020  001334          BIT     #BIT4,@#DSKADR  ;WAS THIS WRITE SURFACE "1"?
 1972  005132  001006                          BNE     3$              ;IF YES BRANCH
 1973  005134  052737  000020  001334          BIS     #BIT4,@#DSKADR  ;SET SURFACE ONE BIT
 1974  005142  105137  001362                  COMB    @#PATTRN        ;MAKE IT SURFACE ONE DATA
 1975  005146  000753                          BR      2$              ;RELOAD REGISTERS AND EXECUTE
 1976  005150  042737  000020  001334  3$:     BIC     #BIT4,@#DSKADR  ;SET UP FOR SURFACE "0"
 1977  005156  105137  001362                  COMB    @#PATTRN        ;MAKE IT SURFACE 0 DATA
 1978  005162  005202                          INC     R2              ;INC. THRU SELECTED CYL. OFFSET TABLE
 1979  005164  004737  005252                  JSR     PC,CYLOFF       ;GET THE CYLINDER VALUE
 1980  005170  000742                          BR      2$              ;RETURN HERE IF MORE TO READ
 1981  005172  000207                          RTS     PC              ;RETURN HERE IF FINISHED
 1982
 1983                                  ;THIS ROUTINE EXPECTS TO FIND A MASK IN R2, AND FROM THIS MASK
 1984                                  ;BUILDS A TABLE (AT CYLTBL) OF CYLINDER ADDRESS OFFSETS; THE TABLE
 1985                                  ;IS TERMINATED BY A #377
 1986
 1987  005174  010546                  MASK:   MOV     R5,-(SP)        ;SAVE R5
 1988  005176  042702  177400                  BIC     #177400,R2      ;CLR THE UNUSED BITS OF THE MASK
 1989  005202  000240                          NOP                     ;***
 1990  005204  012703  000200                  MOV     #200,R3         ;SET UP THE COMPARE MASK
 1991  005210  005004                          CLR     R4              ;CLR THE INDEX COUNTER
 1992  005212  012705  001274                  MOV     #CYLTBL,R5      ;GET THE CYLINDER TABLE ADDRESS
 1993  005216  030203                  1$:     BIT     R2,R3           ;IS THE MASK BIT SELECTED IN BASE
 1994  005220  001401                          BEQ     2$              ;IF NO BRANCH
```

```
 1995  005222  110425                          MOVB    R4,(R5)+        ;MOVE THE CYLINDER BASE TO THE TABLE
 1996  005224  105204                  2$:     INCB    R4              ;INCREMENT THE BASE
 1997  005226  006003                          ROR     R3              ;ROTATE THE COMPARE MASK
 1998  005230  103372                          BCC     1$              ;IF NOT DONE GO BACK
 1999  005232  112715  000377                  MOVB    #377,(R5)       ;IF DONE LOAD FINISH FLAG
 2000  005236  012605                          MOV     (SP)+,R5        ;RESTORE R5
 2001  005240  000207                          RTS     PC              ;RETURN TO CALLER
 2002
 2003                                  ;THIS ROUTINE FORMS A CYLINDER ADDRESS FOR BOTH THE READ AND WRITE ROUTINES
 2004                                  ;WHEN THE BASE TABLE IS FULLY INDEXED IT RETURNS TO PC+2 OF CALLER
 2005
 2006  005242  012703  001266          CYLADR: MOV     #BASE,R3        ;GET THE CYLINDER TABLE ADDRESS
 2007  005246  012702  001274          CYLAD2: MOV     #CYLTBL,R2      ;GET THE SELECTED CYL BASE ADDR.
 2008  005252  111204                  CYLOFF: MOVB    (R2),R4         ;GET THE SELECTED CYL VALUE TO R4
 2009  005254  000240                          NOP                     ;***
 2010  005256  122704  000377                  CMPB    #377,R4         ;IS IT THE TABLE TERMINATOR?
 2011  005262  001416                          BEQ     BASINC          ;IF YES BRANCH
 2012  005264  005046                          CLR     -(SP)           ;INSURE CLEAN WORD
 2013  005266  111316                          MOVB    (R3),(SP)       ;GET THE CYL ADDRESS ON THE STACK
 2014  005270  062604                          ADD     (SP)+,R4        ;AND IT TO THE SELECTED OFFSET
 2015  005272  006104                          ROL     R4              ;SHIFT THIS RESULT
 2016  005274  006104                          ROL     R4              ;TO ALIGN THE NEWLY FORMED
 2017  005276  006104                          ROL     R4              ;CYLINDER ADDRESS WITH BITS
 2018  005300  006104                          ROL     R4              ;5 THRU 12 OF RKDA AND
 2019  005302  006104                          ROL     R4              ;STORE THIS IN DSKADR
 2020  005304  042737  017777  001334          BIC     #017777,@#DSKADR;CLEAR ALL BUT DRIVE NUMBER
 2021  005312  050437  001334                  BIS     R4,@#DSKADR     ;PUT IT IN DSKADR
 2022  005316  000207                          RTS     PC
 2023  005320  005203                  BASINC: INC     R3              ;PICK UP ADDRESS OF NEXT BASE CYL.
 2024  005322  000240                          NOP                     ;***
 2025  005324  022703  001274                  CMP     #CYLTBL,R3      ;ARE YOU FINISHED?
 2026  005330  001401                          BEQ     RETRN3          ;IF YES, BRANCH
 2027  005332  000745                          BR      CYLAD2          ;NO GO BACK
 2028  005334  062716  000002          RETRN3: ADD     #2,(SP)         ;SET-UP FOR PC+2
 2029  005340  000207                          RTS     PC              ;RETURN
 2030
 2031                                  ;ROUTINE TO PERFORM INDICATED FUNCTION AND CHECK FOR
 2032                                  ;DONE AND ERRORS
 2033
 2034  005342  005037  001346          EXECUT: CLR     @#TIMR          ;CLEAR THE TIMER
 2035  005346  000240                          NOP                     ;HALT HERE TO CHECK COMMAND ABOUT TO BE EXECUTED
 2036  005350  105037  001323                  CLRB    @#TIMR2         ;CLEAR SECOND TIMER
 2037  005354  013777  001334  174014          MOV     @#DSKADR,@RKDA  ;LOAD THE DISK ADDRESS REGISTER
 2038  005362  013777  001336  174004          MOV     @#BUSADR,@RKBA  ;LOAD THE BUS ADDRESS REGISTER
 2039  005370  013777  001340  173774          MOV     @#WRDCNT,@RKWC  ;LOAD THE WORD COUNT REGISTER
 2040  005376  013777  001332  173764          MOV     @#CONTRL,@RKCS  ;LOAD THE CONTROL REGISTER
 2041  005404  004737  005554                  JSR     PC,SMTME        ;KILL TIME FOR RK11-C
 2042  005410  105777  173754          CHECK1: TSTB    @RKCS           ;IS CONTROL READY SET
 2043  005414  100011                          BPL     TIME            ;IF NO BRANCH
 2044  005416  004737  006042                  JSR     PC,ERRCHK
 2045  005422  005737  001360                  TST     @#ERRFLG        ;ERROR?
 2046  005426  001403                          BEQ     1$              ;IF NO BRANCH
 2047  005430  005037  001360                  CLR     @#ERRFLG        ;CLEAR THE FLAG
 2048  005434  000742                          BR      EXECUT          ;TRY AGAIN
 2049  005436  000207                  1$:     RTS     PC
 2050  005440  005237  001346          TIME:   INC     @#TIMR
```

```
2051  005444  000240                          NOP                     ;***
2052  005446  00..60                          BNE     CHECK1
2053  005450  105737  001323                  TSTB    @#TIMR2         ;SECOND TIMEOUT?
2054  005454  001003                          BNE     1$              ;IF YES BRANCH
2055  005456  105237  001323                  INCB    @#TIMR2         ;INDICATE SECOND TIMEOUT
2056  005462  000752                          BR      CHECK1          ;GO BACK
2057  005464  004737  004464          1$:     JSR     PC,INITIL
2058  005470  104401  005476                  TYPE    ,65$            ;;TYPE ASCIZ STRING
2059  005474  000426                          BR      64$             ;;GET OVER THE ASCIZ
2060                                  ;;65$:  .ASCIZ  <15><12>/TIMED OUT ON OPERATION RETRY IN PROGRESS/
2061  005552                          64$:
2062  005552  000673                          BR      EXECUT
2063  005554  012737  000500  001346  SMTME:  MOV     #500,@#TIMR
2064  005562  005337  001346          1$:     DEC     @#TIMR
2065  005566  001375                          BNE     1$
2066  005570  000207                          RTS     PC
2067
2068                                  ;THIS ROUTINE CHECKS TO SEE IF A DRIVE IS ACTIVE FROM A
2069                                  ;WRITE OPERATION, IT GETS THE READ MASK TO R3, AND THE
2070                                  ;EXPECTED DATA PATTERN TO "PATTERN", AND THEN CALLS ADDRESS
2071                                  ;CONTROL.
2072
2073  005572  010046                  RDLINK: MOV     R0,-(SP)        ;SAVE R0
2074  005574  000240                          NOP                     ;***
2075  005576  152737  000377  001316          BISB    #377,@#COMND    ;INDICATE READ OPERATION
2076  005604  012701  001166                  MOV     #LOGA,R1        ;GET THE TABLE ADDRESS TO R1
2077  005610  012705  001305                  MOV     #SECTBL,R5      ;GET THE SECTOR TABLE ADDRESS
2078  005614  000405                          BR      RD2             ;SKIP OVER THE INDEX, FIRST PASS
2079  005616  062701  000002          RD1:    ADD     #2,R1           ;ADD 2 TO THE ADDRESS
2080  005622  022701  001206                  CMP     #DRV0,R1        ;ARE YOU THRU THE ENTIRE TABLE
2081  005626  001503                          BEQ     EXIT2           ;IF YES EXIT
2082  005630  005711                  RD2:    TST     (R1)            ;IS THE DRIVE ACTIVE
2083  005632  100001                          BPL     RD3             ;IF YES BRANCH
2084  005634  000770                          BR      RD1             ;IF NO GET NEXT WORD
2085  005636  011100                  RD3:    MOV     (R1),R0         ;GET THE ACTIVE WORD TO R0
2086  005640  010002                          MOV     R0,R2           ;COPY THE WORD
2087  005642  000300                          SWAB    R0              ;GET THE DRIVE # TO THE LOW BYTE
2088  005644  042700  177770                  BIC     #177770,R0      ;CLR ALL BUT THE DRIVE #
2089  005650  110037  001317                  MOVB    R0,@#WRTNBY     ;SAVE THE DRIVE #
2090  005654  006100                          ROL     R0              ;MAKE IT AN INDEX
2091  005656  016037  001206  001362          MOV     DRV0(R0),@#PATTRN ;PICK UP THE DATA PATTERN
2092  005664  004737  005174                  JSR     PC,MASK         ;CALL THE MASK SUBROUTINE
2093  005670  004737  005242                  JSR     PC,CYLADR       ;GO FORM A CYLINDER ADDRESS
2094  005674  000401                          BR      RD4             ;RETURN HERE IF NOT LAST BASE ADDR.
2095  005676  000747                          BR      RD1             ;IF LAST BASE ADDRESS, RETURN HERE
2096  005700  053737  001352  001334  RD4:    BIS     @#DSKTMP,@#DSKADR ;SET THE DRIVE # BITS IN DISK ADDR.
2097  005706  152537  001334          RD5:    BISB    (R5)+,@#DSKADR  ;SET THE SECTOR BITS IN DISK ADDR.
2098  005712  012737  007316  001336  RD6:    MOV     #RDBUFF,@#BUSADR ;GET THE BUFFER ADDR.
2099  005720  000240                          NOP                     ;***
2100  005722  013737  001344  001340          MOV     @#SECCNT,@#WRDCNT ;GET THE WORD COUNT
2101  005730  013737  001356  001332          MOV     @#READCS,@#CONTRL ;GET THE READ CONTROL WORD
2102  005736  004737  005342                  JSR     PC,EXECUT       ;DO THE READ
2103  005742  004737  006410                  JSR     PC,RDCHK        ;CHECK THE DATA
2104  005746  042737  000017  001334          BIC     #17,@#DSKADR    ;CLR THE SECTOR BITS IN DISK ADDR.
2105  005754  022705  001311                  CMP     #DRCNT1,R5      ;WAS THIS THE LAST SECTOR?
2106  005760  001352                          BNE     RD5             ;IF NO GO BACK
```

```
2107  005762  012705  001305                MOV     #SECTBL,R5      ;IF YES RESET SECTOR POINTER
2108  005766  032737  000020  001334        BIT     #BIT4,@#DSKADR  ;WAS IT SURFACE "1" THAT WAS READ?
2109  005774  001006                        BNE     RD7             ;IF YES, BRANCH
2110  005776  052737  000020  001334        BIS     #BIT4,@#DSKADR  ;NO, SET SURFACE "1" BIT
2111  006004  105137  001362                COMB    @#PATTRN        ;MAKE HEAD "1" PATTERN
2112  006010  000736                        BR      RD5             ;GO BACK AND EXECUTE
2113  006012  042737  000020  001334  RD7:  BIC     #BIT4,@#DSKADR  ;CLEAR THE SURFACE BIT
2114  006020  105137  001362                COMB    @#PATTRN        ;MAKE IT SURFACE 0 DATA
2115  006024  005202                        INC     R2              ;INCREMENT THE SELECTED CYL TABLE POINTER
2116  006026  004737  005252                JSR     PC,CYLOFF       ;GO FORM NEXT ADDRESS
2117  006032  000722                        BR      RD4             ;IF HERE IT IS NOT THE LAST BASE ADDR
2118  006034  000670                        BR      RD1             ;IF HERE GET NEXT WORD-DRIVE FINISHED
2119
2120  006036  012600          EXIT2:  MOV     (SP)+,R0        ;RESTORE R0
2121  006040  000207                  RTS     PC              ;RETURN TO MAIN LINE CODE
2122
2123                          ;THE ERROR CHECK ROUTINE CHECKS IF AN ERROR OCCURRED
2124                          ;ON WRITING OR READING.
2125
2126  006042  032777  140000  173320  ERRCHK: BIT     #140000,@RKCS   ;HARD ERROR OR ERROR SET?
2127  006050  000240                  NOP             ;HALT HERE TO EXAMINE ERROR REG.,ECT.
2128  006052  001420                  BEQ     TSTSN1  ;IF NO, GO TEST 'SIN' BIT
2129  006054  012777  000001  173306        MOV     #1,@RKCS        ;IF YES, ISSUE CNTROL RESET + GO
2130  006062  004737  005554                JSR     PC,SMTME
2131  006066  012777  177777  173264        MOV     #-1,@ERRFLG     ;FLAG AN ERROR (PREVENT UPDATE OF ADDR)
2132  006074  105777  173270          1$:   TSTB    @RKCS           ;CNTROL READY BIT SET (FROM CNTROL RESET)
2133  006100  100375                  BPL     1$              ;IF NO WAIT.  (IF HUNG HERE RUN STATIC)
2134  006102  105337  001321                DECB    @#ECNT          ;DECREMENT ERROR COUNTER
2135  006106  001002                  BNE     TSTSN1          ;HAVE ERROR BITS SET 5 TIMES?
2136  006110  000137  006172                JMP     @#RESTRT        ;IF HERE 5 ERRORS HAVE OCCURRED
2137  006114  032777  001000  173242  TSTSN1: BIT     #1000,@RKDS     ;SEEK INCOMPLETE SET?
2138  006122  000240                  NOP                     ;***
2139  006124  001530                  BEQ     RETRN2          ;BRANCH IF NO
2140  006126  012777  000015  173234        MOV     #15,@RKCS       ;IF YES, ISSUE DRIVE RESET, GO
2141  006134  012777  177777  173216        MOV     #-1,@ERRFLG     ;FLAG AN ERROR (PREVENT UPDATE OF ADDR)
2142  006142  004737  005554                JSR     PC,SMTME
2143  006146  105777  173216          2$:   TSTB    @RKCS
2144  006152  100375                  BPL     2$
2145  006154  032777  000100  173202        BIT     #100,@RKDS      ;"R/W/S READY" BIT SET?
2146  006162  001771                  BEQ     2$              ;IF NO WAIT! (IF HUNG HERE RUN STATIC)
2147  006164  105337  001322                DECB    @#CNTSIN        ;DECREMENT SEEK INCOMPLETE COUNTER
2148  006170  001106                  BNE     RETRN2          ;IF 3 'SIN' ERRORS FALL THROUGH
2149  006172  105737  001321  RESTRT: TSTB    @#ECNT
2150  006176  000240                  NOP                     ;***
2151  006200  001421                  BEQ     1$
2152  006202  104401  006210                TYPE    ,65$            ;;TYPE ASCIZ STRING
2153  006206  000415                  BR      64$             ;;GET OVER THE ASCIZ
2154                          ;;65$:  .ASCIZ  <15><12>/3 'SIN' ERRORS OCCURRED/
2155  006242                  64$:
2156  006242  000415                  BR      2$
2157  006244                  1$:
2158  006244  104401  006252                TYPE    ,67$            ;;TYPE ASCIZ STRING
2159  006250  000412                  BR      66$             ;;GET OVER THE ASCIZ
2160                          ;;67$:  .ASCIZ  <15><12>/5 ERRORS OCCURRED/
2161  006276                  66$:
2162  006276                  2$:
```

```
 2163  006276  104401  006304                    TYPE    ,69$            ;;TYPE ASCIZ STRING
 2164  006302  000422                            BR      68$             ;;GET OVER THE ASCIZ
 2165                                    ;;69$:  .ASCIZ  / DRIVE DECLARED DOWN!! NOT TESTED !/
 2166  006350                            68$:
 2167  006350  105737  001316                    TSTB    @#COMND         ;TEST THE COMMAND
 2168  006354  100006                            BPL     3$              ;IF WRITE, BPANCH
 2169  006356  062706  000004                    ADD     #4,SP           ;POINT TO SAVE RO
 2170  006362  012600                            MOV     (SP)+,RO        ;GET RO BACK
 2171  006364  052710  100000                    BIS     #BIT15,(RO)     ;SET THE DOWN BIT
 2172  006370  000207                            RTS     PC              ;RETURN TO MAIN CODE
 2173  006372  052710  100000            3$:     BIS     #BIT15,(RO)     ;SET THE DOWN BIT
 2174  006376  012706  001100                    MOV     #STACK,SP       ;RESTORE THE STACK
 2175  006402  000137  003264                    JMP     @#GO1
 2176  006406  000207                    RETRN2: RTS     PC
 2177
 2178                                    ;THIS ROUTINE CHECKS A SECTORS WORTH OF DATA ON A READ OPERATION
 2179                                    ;IT ALLOWS 5 ERROR PRINTOUTS PER SECTOR
 2180
 2181  006410  010446                    RDCHK:  MOV     R4,-(SP)        ;SAVE   R4
 2182  006412  010546                            MOV     R5,-(SP)        ;SAVE R5 FOR RDLINK
 2183  006414  105037  001320                    CLRB    @#HDRFLG        ;CLEAR THE PRINT HEADER FLAG
 2184  006420  000240                            NOP                     ;***
 2185  006422  012737  000005  001350            MOV     #5,@#CHKCNT     ;PUT ERROR COUNT IN CHECK COUNT
 2186  006430  012704  007316                    MOV     #ROBUFF,R4      ;GET THE TABLE ADDRESS TO R4
 2187  006434  013705  001362                    MOV     @#PATTRN,R5     ;GET THE EXPECTED DATA TO R5
 2188  006440  020524                    1$:     CMP     R5,(R4)+        ;ARE THEY THE SAME
 2189  006442  001515                            BEQ     3$              ;IF YES BRANCH
 2190  006444  105737  001320                    TSTB    @#HDRFLG        ;IS THE HEADER FLAG CLEAR
 2191  006450  001046                            BNE     2$              ;IF NO BRANCH
 2192  006452  104401  006460                    TYPE    ,65$            ;;TYPE ASCIZ STRING
 2193  006456  000420                            BR      64$             ;;GET OVER THE ASCIZ
 2194                                    ;;65$:  .ASCIZ  <15><12>/ERROR! DATA WRITTEN BY DRIVE /
 2195  006520                            64$:
 2196  006520  152737  000377  001320            BISB    #377,@#HDRFLG   ;SET THE HEADER FLAG
 2197  006526  113746  001317                    MOVB    @#WRTNBY,-(SP)  ;PICK UP THE DRIVE # THAT WROTE
 2198  006532  104403                            TYPOS
 2199  006534     001                            .BYTE   1
 2200  006535     000                            .BYTE   0
 2201  006536  104401  006544                    TYPE    ,67$            ;;TYPE ASCIZ STRING
 2202  006542  000411                            BR      66$             ;;GET OVER THE ASCIZ
 2203                                    ;;67$:  .ASCIZ  / CANNOT BE READ./
 2204  006566                            66$:
 2205  006566                            2$:
 2206  006566  104401  006574                    TYPE    ,69$            ;;TYPE ASCIZ STRING
 2207  006572  000405                            BR      68$             ;;GET OVER THE ASCIZ
 2208                                    ;;69$:  .ASCIZ  <15><12>/  ADDR=/
 2209  006606                            68$:
 2210  006606  013746  001334                    MOV     @#DSKADR,-(SP)  ;PICK UP THE ADDRESS THAT FAILED
 2211  006612  104403                            TYPOS
 2212  006614     006                            .BYTE   6
 2213  006615     001                            .BYTE   1
 2214  006616  104401  006624                    TYPE    ,71$            ;;TYPE ASCIZ STRING
 2215  006622  000405                            BR      70$             ;;GET OVER THE ASCIZ
 2216                                    ;;71$:  .ASCIZ  /  EXPCTD=/
 2217  006636                            70$:
 2218  006636  011546                            MOV     R5,-(SP)        ;PICK UP THE EXPECTED DATA (GOOD)
```

```
 2219  006640  104404                          TYPON
 2220  006642  104401  006650                  TYPE    ,73$            ;;TYPE ASCIZ STRING
 2221  006646  000405                          BR      72$             ;;GET OVER THE ASCIZ
 2222                                  ;;73$:  .ASCIZ  /  RECVD=/
 2223  006662                                72$:
 2224  006662  016446  177776                  MOV     -2(R4),-(SP)    ;PICK UP THE RECEIVED DATA (BAD)
 2225  006666  104404                          TYPON
 2226  006670  005337  001350                  DEC     @#CHKCNT        ;DECREMENT THE CHECK COUNT
 2227  006674  001403                          BEQ     4$              ;IF ZERO, BRANCH
 2228  006676  022704  010316          3$:     CMP     #MANSEL,R4      ;DONE ALL CHECKS?
 2229  006702  001256                          BNE     1$              ;IF NO, GO BACK
 2230  006704  012605                  4$:     MOV     (SP)+,R5        ;RESTORE R5
 2231  006706  012604                          MOV     (SP)+,R4        ;RESTORE R4
 2232  006710  000207                          RTS     PC              ;RETURN TO CALLER
 2233
 2234                                  ;THIS ROUTINE BUILDS A TABLE OF PARAMETERS TO PASS TO THE SECOND SYSTEM
 2235                                  ;IT PACKS THE INFO FOR TWO DRIVES INTO ONE WORD
 2236
 2237  006712  005003                  SECONE: CLR     R3              ;CLEAR THE WORD COUNTER
 2238  006714  000240                          NOP                     ;***
 2239  006716  012702  001256                  MOV     #MSKTBL,R2
 2240  006722  005042                  6$:     CLR     -(R2)
 2241  006724  022702  001246                  CMP     #PASTBL,R2
 2242  006730  001374                          BNE     6$
 2243  006732  012700  001166                  MOV     #LOGA,R0        ;GET THE ACTIVE TABLE ADDRESS
 2244  006736  012001                  1$:     MOV     (R0)+,R1        ;PICK UP THE WORD
 2245  006740  000301                          SWAB    R1              ;GET THE DRIVE # TO THE LOW BYTE
 2246  006742  042701  177400                  BIC     #177400,R1      ;CLEAR THE UNWANTED BITS
 2247  006746  106101                          ROLB    R1              ;ROTATE THE BYTE, WAS DOWN SET?
 2248  006750  103002                          BCC     2$              ;IF NO BRANCH
 2249  006752  052701  000001                  BIS     #BIT0,R1        ;SHOW THE DRIVE AS DOWN IN THE TABLE
 2250  006756  105701                  2$:     TSTB    R1              ;IS THIS THE SYSTEM #2 DRIVE?
 2251  006760  100404                          BMI     3$              ;BRANCH IF YES
 2252  006762  142701  000360                  BICB    #360,R1         ;CLEAR THE UNUSED BITS
 2253  006766  110122                          MOVB    R1,(R2)+        ;GET THIS # TO THE PASS TABLE
 2254  006770  000762                          BR      1$              ;GET THE NEXT WORD FROM ACTIVE TABLE
 2255  006772  110112                  3$:     MOVB    R1,(R2)         ;GET THE LAST DRIVE TO THE PASS TABLE
 2256  006774  012702  001246                  MOV     #PASTBL,R2      ;RESTORE THE TABLE POINTER
 2257  007000  104401  007006                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 2258  007004  000425                          BR      64$             ;;GET OVER THE ASCIZ
 2259                                  ;;65$:  .ASCIZ  <15><12>/LOAD AND START ADDRESS 210 ON SYSTEM #2/
 2260  007060                                64$:
 2261  007060  104401  007066                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 2262  007064  000424                          BR      66$             ;;GET OVER THE ASCIZ
 2263                                  ;;67$:  .ASCIZ  <15><12>/AND TYPE THE BELOW WHEN ASKED FOR IT./
 2264  007136                                66$:
 2265  007136  005203                  4$:     INC     R3              ;INCREMENT THE WORD COUNTER
 2266  007140  104401  007146                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 2267  007144  000405                          BR      68$             ;;GET OVER THE ASCIZ
 2268                                  ;;69$:  .ASCIZ  <15><12>/WORD /
 2269  007160                                68$:
 2270  007160  010346                          MOV     R3,-(SP)        ;GET THE WORD COUNT ON THE STACK
 2271  007162  104403                          TYPOS
 2272  007164     006                          .BYTE   6
 2273  007165     000                          .BYTE   0
 2274  007166  104401  007174                  TYPE    ,71$            ;;TYPE ASCIZ STRING
```

```
 2275  007172  000401                          BR      70$             ;;GET OVER THE ASCIZ
 2276                                  ;;71$:  .ASCIZ  /=/
 2277  007176                          70$:
 2278  007176  012246                          MOV     (R2)+,-(SP)     ;GET THE FIRST TO THE STACK
 2279  007200  104403                          TYPOS
 2280  007202     006                          .BYTE   6
 2281  007203     001                          .BYTE   1
 2282  007204  032762  100000  177776          BIT     #BIT15,-2(R2)   ;WAS THIS THE TABLE TERMINATOR
 2283  007212  001004                          BNE     5$              ;BRANCH IF YES
 2284  007214  032762  000200  177776          BIT     #BIT7,-2(R2)    ;TERMINATOR?
 2285  007222  001745                          BEQ     4$              ;IF NO BRANCH
 2286  007224  005740                  5$:     TST     -(R0)
 2287  007226  000000                          HALT
 2288
 2289  007230                          RETFR2:
 2290  007230  104401  007236                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 2291  007234  000404                          BR      64$             ;;GET OVER THE ASCIZ
 2292                                  ;;65$:  .ASCIZ  <15><12>/WORD=/
 2293  007246                          64$:
 2294  007246  104407                          RDOCT
 2295  007250  012602                          MOV     (SP)+,R2        ;GET THE WORD FROM SYSTEM 2 TO TABLE
 2296  007252  042702  177400                  BIC     #177400,R2
 2297  007256  012704  001166                  MOV     #LOGA,R4        ;SET POINTER LOOK FOR FIRST "UP" DRIVE
 2298  007262  005724                  1$:     TST     (R4)+           ;DRIVE UP?
 2299  007264  100776                          BMI     1$              ;IF NO BRANCH
 2300  007266  010437  001100                  MOV     R4,@#$PASS
 2301  007272  014401                          MOV     -(R4),R1
 2302  007274  000240                          NOP                     ;***
 2303  007276  004737  004164                  JSR     PC,LDFLG        ;CALL DO2+
 2304  007302  004737  005572                  JSR     PC,RDLINK       ;CALL READ CHECK
 2305  007306  013700  001100                  MOV     @#$PASS,R0
 2306  007312  000137  003332                  JMP     @#EXTFR2        ;GO TO END OF TEST
 2307
 2308  007316  000400                  RDBUFF: .BLKW   400
 2309
 2310  010316  000240                  MANSEL: NOP                     ;TABLE TERMINATOR
 2311
 2312
 2313
 2314
 2315
 2316  010320                          BADONE:
 2317  010320  104401  010326                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 2318  010324  000401                          BR      64$             ;;GET OVER THE ASCIZ
 2319                                  ;;65$:  .ASCIZ  /?/
 2320  010330                          64$:
 2321
 2322
 2323                                          .SBTTL  OSCILLATING SEEK ROUTINE
 2324
 2325  010330                          SECT.2:
 2326  010330  104401  010336                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 2327  010334  000416                          BR      64$             ;;GET OVER THE ASCIZ
 2328                                  ;;65$:  .ASCIZ  <15><12>/OSCILLATING SEEK PACKAGE
 2329  010372                          64$:
 2330
```

```
2331  010372  012700  020436                    MOV     #DRIVO,RO       ;FIND OUT WHICH DRIVES ARE
2332  010376  005001                            CLR     R1              ;PRESENT AND PUT THE
2333  010400  010177  170772          1$:       MOV     R1,@RKDA        ;DRIVE #'S IN A TABLE STARTING
2334  010404  105777  170754                    TSTB    @RKDS           ;AT 'DRIVO'. BITS 15-13 CONTAINS
2335  010410  100001                            BPL     2$              ;THE DRIVE #
2336  010412  010120                            MOV     R1,(RO)+
2337  010414  062701  020000          2$:       ADD     #20000,R1
2338  010420  001367                            BNE     1$
2339  010422  012710  177777                    MOV     #-1,(RO)        ;SET THE TERMINATOR TO THE TABLE
2340
2341  010426  013702  001376          INIT.2:   MOV     RKDA,R2
2342  010432  012777  000001  170730            MOV     #1,@RKCS        ;ISSUE CONTROL RESET + GO !
2343  010440  004737  005554                    JSR     PC,SMTME
2344  010444  105777  170720          1$:       TSTB    @RKCS           ;DID CONTROL READY SET?
2345  010450  100375                            BPL     1$              ;IF NO WAIT! (IF HUNG RUN STATIC)
2346  010452  012700  020436                    MOV     #DRIVO,RO
2347  010456  012077  170714          3$:       MOV     (RO)+,@RKDA
2348  010462  012777  000015  170700            MOV     #15,@RKCS       ;ISSUE DRIVE RESET + GO!
2349  010470  004737  005554                    JSR     PC,SMTME
2350  010474  105777  170670          2$:       TSTB    @RKCS
2351  010500  100375                            BPL     2$
2352  010502  022710  177777                    CMP     #-1,(RO)
2353  010506  001363                            BNE     3$
2354
2355  010510  104401  010516                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2356  010514  000432                            BR      64$             ;;GET OVER THE ASCIZ
2357                                  ;;65$:    .ASCIZ  <15><12>/SET SW0 TO SW7 TO SELECT DRIVES FOR TEST AND CONT/
2358  010602                          64$:
2359  010602  104401  010610                    TYPE    ,67$            ;;TYPE ASCIZ STRING
2360  010606  000426                            BR      66$             ;;GET OVER THE ASCIZ
2361                                  ;;67$:    .ASCIZ  <15><12>/RESET SW0 TO SW7 TO TEST ALL AVAIL DRIVES/
2362  010664                          66$:
2363  010664  000000                            HALT
2364  010666  117704  170246                    MOVB    @SWR,R4         ;SWO TO SW7 TO R4
2365  010672  001413                            BEQ     4$              ;NONE SET, SO TEST ALL
2366  010674  012700  020436                    MOV     #DRIVO,RO       ;TABLE TO STORE DRIVE ADDRS
2367  010700  005001                            CLR     R1              ;ADDR OF DRIVE
2368  010702  006004          6$:       ROR     R4                      ;NEXT SWITCH TO CARRY
2369  010704  103001                            BCC     5$              ;SWITCH NOT SET
2370  010706  010120                            MOV     R1,(RO)+        ;SWITCH SET, SO MOVE ADDR TO TABLE
2371  010710  062701  020000          5$:       ADD     #20000,R1       ;ADDR OF NEXT DRIVE
2372  010714  001372                            BNE     6$              ;ALL DONE WHEN ZERO
2373  010716  012720  177777                    MOV     #-1,(RO)+       ;TABLE TERMINATOR
2374  010722  105737  001326          4$:       TSTB    OSPFLG          ;TYPED ONCE?
2375  010726  001165                            BNE     RWSRDY          ;IF YES, DONT RETYPE
2376  010730  104401  010736                    TYPE    ,69$            ;;TYPE ASCIZ STRING
2377  010734  000432                            BR      68$             ;;GET OVER THE ASCIZ
2378                                  ;;69$:    .ASCIZ  <15><12>/TOGGLE THE "FIRST CYLINDER ADDRESS" (OUTER LIMIT)/
2379  011022                          68$:
2380  011022  104401  011030                    TYPE    ,71$            ;;TYPE ASCIZ STRING
2381  011026  000441                            BR      70$             ;;GET OVER THE ASCIZ
2382                                  ;;71$:    .ASCIZ  <15><12>/INTO THE LOW BYTE (BITO-7) OF THE SWITCH REGISTER AND THE "LAST
2383  011132                          70$:
2384  011132  104401  011140                    TYPE    ,73$            ;;TYPE ASCIZ STRING
2385  011136  000430                            BR      72$             ;;GET OVER THE ASCIZ
2386                                  ;;73$:    .ASCIZ  <15><12>/CYLINDER ADDRESS" (INNER LIMIT) INTO THE HIGH
```

```
2387  011220                          72$:
2388  011220  104401  011226                TYPE    ,75$            ;;TYPE ASCIZ STRING
2389  011224  000424                        BR      74$             ;;GET OVER THE ASCIZ
2390                                ;;75$:  .ASCIZ  <15><12>/       BYTE (BITS-15), THEN PRESS CONTINUE./
2391  011276                        74$:
2392
2393  011276  105237  001326              INCB    OSPFLG
2394  011302  032777  000100  170054 RWSRDY: BIT    #100,@RKDS      ;"READ/WRITE/SEEK READY" BIT SET?
2395  011310  001774                        BEQ     RWSRDY          ;IF NO WAIT! (IF HUNG RUN STATIC)
2396
2397  011312  000000                TRYAGN: HALT
2398
2399  011314  012777  000001  170046 CONTIN: MOV     #1,@RKCS        ;CONTROL RESET
2400  011322  105777  170042              TSTB    @RKCS
2401  011326  100375                        BPL     .-4
2402  011330  013705  001140              MOV     SWR,R5          ;GET THE ADDRESS OF THE SWITCH REG. TO R5
2403  011334  112501                1$:     MOVB    (R5)+,R1        ;GET A BYTE TO R1
2404  011336  042701  177400              BIC     #177400,R1      ;CLEAR THE UNUSED BITS
2405  011342  022701  000312              CMP     #312,R1         ;IS ADDRESS LEGAL?
2406  011346  100034                        BPL     2$              ;BRANCH IF YES
2407  011350  104401  011356              TYPE    ,65$            ;;TYPE ASCIZ STRING
2408  011354  000430                        BR      64$             ;;GET OVER THE ASCIZ
2409                                ;;65$:  .ASCIZ  <15><12>/INVALID ADDRESS IN SWITCH REGISTER TRY AGAIN/
2410  011436                        64$:
2411  011436  000725                        BR      TRYAGN          ;GO BACK FOR NEW PARAMETERS
2412  011440  006101                2$:     ROL     R1              ;ROTATING THIS REGISTER
2413  011442  006101                        ROL     R1              ;MAKES THE BYTE FROM THE
2414  011444  006101                        ROL     R1              ;SWITCH REGISTER LINE UP
2415  011446  006101                        ROL     R1              ;WITH THE CYLINDER BITS OF
2416  011450  006101                        ROL     R1              ;THE RKDA REGISTER.
2417  011452  042701  160037              BIC     #160037,R1      ;CLEAR THE UNUSED BITS
2418  011456  032705  000001              BIT     #1,R5           ;IS THIS THE LOW BYTE?
2419  011462  001003                        BNE     3$              ;BRANCH IF YES
2420  011464  010137  001402              MOV     R1,@#SEEKI      ;STORE THE INNER LIMIT OF THE SEEK
2421  011470  000403                        BR      SEKSET          ;START SEEKS
2422  011472  010137  001404          3$:   MOV     R1,@#SEEKO      ;STORE THE OUTER LIMIT OF THE SEEK
2423  011476  000716                        BR      1$              ;GO BACK AND GET HIGH BYTE.
2424
2425  011500  012703  000050        SEKSET: MOV     #50,R3          ;GET THE NUMBER OF SEEK CYCLES TO R3
2426  011504  013704  001404              MOV     @#SEEKO,R4      ;ADD THE OUTER LIMIT CYLINDER ADDRESS
2427  011510  013705  001402              MOV     @#SEEKI,R5      ;ADD THE INNER LIMIT CYLINDER ADDRESS
2428
2429  011514  012700  020436        LDSEEK: MOV     #DRIVO,R0       ;INITIALIZE POINTER
2430  011520  010477  167652          5$:   MOV     R4,@RKDA        ;GET INNER LIMIT CYL ADRES
2431  011524  052077  167646              BIS     (R0)+,@RKDA     ;SET DRIVE ADRES
2432  011530  012777  000011  167632      MOV     #11,@RKCS       ;ISSUE SEEK+GO! (FOR INNER LIMIT)
2433  011536  004737  005554              JSR     PC,$MTME
2434  011542  105777  167622          3$:   TSTB    @RKCS
2435  011546  100375                        BPL     3$
2436
2437  011550  022710  177777              CMP     #-1,(R0)        ;ALL DRIVES DONE?
2438  011554  001361                        BNE     5$              ;NO
2439  011556  012700  020436              MOV     #DRIVO,R0
2440  011562  012077  167610          6$:   MOV     (R0)+,@RKDA     ;ADRES THE DRIVE
2441  011566  032777  000100  167570  7$:   BIT     #RWS,@RKDS      ;SEEK DONE?
2442  011574  001774                        BEQ     7$              ;NO, WAIT
```

```
2443  011576  022710  177777                    CMP     #-1,(R0)        ;ALL DRIVES DONE?
2444  011602  001367                            BNE     6$              ;NO
2445
2446  011604  012700  020436                    MOV     #DRIV0,R0
2447  011610  010577  167562            8$:     MOV     R5,@RKDA        ;SET OUTER CYL ADRES
2448  011614  052077  167556                    BIS     (R0)+,@RKDA     ;SET DRIVE ADRES
2449  011620  012777  000011  167542            MOV     #11,@RKCS       ;ISSUE SEEK+GO! (FOR OUTER LIMIT)
2450  011626  004737  005554                    JSR     PC,$MTME
2451  011632  105777  167532            9$:     TSTB    @RKCS
2452  011636  100375                            BPL     9$
2453
2454  011640  022710  177777                    CMP     #-1,(R0)        ;ALL DONE?
2455  011644  001361                            BNE     8$              ;NO
2456
2457  011646  012700  020436                    MOV     #DRIV0,R0
2458  011652  012077  167520            10$:    MOV     (R0)+,@RKDA     ;SET DRIVE ADRES
2459  011656  032777  000100  167500    11$:    BIT     #RWS,@RKDS              ;SEEK DONE?
2460  011664  001774                            BEQ     11$
2461  011666  022710  177777                    CMP     #-1,(R0)        ;ALL DONE?
2462  011672  001367                            BNE     10$
2463  011674  005303                            DEC     R3              ;DONE 50 SEEK CYCLES (100 SEEKS)
2464  011676  001306                            BNE     LDSEEK          ;IF NO BRANCH (KEEP CYCLING)
2465  011700  000605                            BR      CONTIN          ;CHECK SWR FOR CHANGE AND CONTINUE
2466
2467
2468
2469
2470  011702                            BAD.IN:
2471  011702  104401  011710                            TYPE    ,65$            ;;TYPE ASCIZ STRING
2472  011706  000401                            BR      64$             ;;GET OVER THE ASCIZ
2473                                    ;;65$:  .ASCIZ  /?/
2474  011712                            64$:
2475
2476                                            .SBTTL  FORMATTER-SURFACE VERIFIER
2477
2478  011712                            SECT.1:
2479  011712  104401  011720                            TYPE    ,65$            ;;TYPE ASCIZ STRING
2480  011716  000441                            BR      64$             ;;GET OVER THE ASCIZ
2481                                    ;;65$:  .ASCIZ  <15><12>/FORMATTER-SURFACE VERIFIER,SET SW REG FOR DRV #'S. PRESS CONT./
2482  012022                            64$:
2483  012022  000000                            HALT                    ;WAIT FOR 'CONTINUE'
2484
2485  012024  012737  000001  020266            MOV     #1,SHFCNT       ;SET SHIFT COUNT
2486  012032  005037  020270                    CLR     DRVCNT          ;CLEAR DRIVE COUNT
2487  012036  033777  020266  167074    S13:    BIT     SHFCNT,@SWR     ;IS THIS SW SET?
2488  012044  001011                            BNE     S10             ;YES FORMAT THIS DRIVE
2489  012046  006337  020266            S11:    ASL     SHFCNT
2490  012052  005237  020270                    INC     DRVCNT
2491  012056  022737  000010  020270            CMP     #10,DRVCNT      ;ALL DONE?
2492  012064  001556                            BEQ     GD1
2493  012066  000763                            BR      S13
2494  012070  013700  020270            S10:    MOV     DRVCNT,R0
2495  012074  104401  001161                    TYPE    ,$CRLF
2496  012100  104401  020272                    TYPE    ,EM1            ;TYPE 'DRIVE'
2497  012104  010046                            MOV     R0,-(SP)        ;TYPE DRIVE #
2498  012106  104402                            TYPOC
```

```
 2499  012110  000300                          SWAB    R0
 2500  012112  006100                          ROL     R0
 2501  012114  006100                          ROL     R0
 2502  012116  006100                          ROL     R0
 2503  012120  006100                          ROL     R0
 2504  012122  006100                          ROL     R0
 2505  012124  010037  001352                  MOV     R0,@#DSKTMP
 2506  012130  012737  000000  001422          MOV     #0,ERRWF
 2507  012136  012737  000000  001424          MOV     #0,ERRRF        ;CLEAR OUT ERROR COUNTS.
 2508  012144  012737  000000  001426          MOV     #0,ERRRFC
 2509  012152  012737  000000  001430          MOV     #0,ERRWCH
 2510  012160  012737  000000  001432          MOV     #0,ERRWCS
 2511  012166  012737  177750  001416  S12:    MOV     #-24.,RWC       ;SET WORD COUNT FOR READ FORMAT.
 2512  012174  012737  164000  001412          MOV     #-6144.,WC      ;SET UP WORD COUNT FOR WRITES.
 2513  012202  012737  000000  001420          MOV     #0,EXTR         ;CLEAR EXTRA BIT FOR 12 SECTOR PACK.
 2514  012210  012701  177772          COMMON: MOV     #-6.,%1         ;SET UP LOOP COUNT FOR THE CLEANER.
 2515  012214  012777  014500  167154  COM:    MOV     #14500,@RKDA    ;SET UP FOR A SEEK TO 202.
 2516  012222  053777  001352  167146          BIS     @#DSKTMP,@RKDA
 2517  012230  105777  167134          1$:     TSTB    @RKCS           ;IS THE CONTROLER READY?
 2518  012234  100375                          BPL     1$              ;NO SO WAIT.
 2519  012236  012777  000011  167124          MOV     #11,@RKCS       ;DO THE SEEK.
 2520  012244  032777  000100  167112  2$:     BIT     #BIT6,@RKDS     ;IS THE SEEK DONE?
 2521  012252  001774                          BEQ     2$              ;NO SO WAIT.
 2522  012254  105777  167110          3$:     TSTB    @RKCS           ;IS CONTROLER READY?
 2523  012260  100375                          BPL     3$              ;NO
 2524  012262  012777  000015  167100          MOV     #15,@RKCS       ;DO A DRIVE RESET.
 2525  012270  032777  000100  167066  4$:     BIT     #BIT6,@RKDS     ;IS DRIVE RESET DONE.
 2526  012276  001774                          BEQ     4$              ;NO
 2527  012300  005201                          INC     %1              ;COUNT THE CLEANER LOOP.
 2528  012302  001344                          BNE     COM             ;MORE TO GO.
 2529  012304  012737  000000  001410          MOV     #0,DA           ;START OUT AT CYL. 0.
 2530  012312  012777  177777  167066  NEXT:   MOV     #177777,@BA     ;PUT ALL ONE'S IN BUFFER.
 2531  012320  004137  012426                  JSR     %1,IO           ;GO DO THE DISK THING.
 2532  012324  012777  000000  167054          MOV     #0,@BA          ;PUT ALL ZERO'S IN BUFFER.
 2533  012332  004137  012426                  JSR     %1,IO           ;GO DO IT AGIAN.
 2534  012336  012777  125252  167042          MOV     #125252,@BA     ;PUT A ALT. PATTERN IN BUFFER.
 2535  012344  004137  012426                  JSR     %1,IO           ;ONCE MORE.
 2536  012350  005177  167032                  COM     @BA             ;COMPLEMENT THE LAST PATTERN.
 2537  012354  004137  012426                  JSR     %1,IO           ;AND AGAIN.
 2538  012360  062737  000040  001410          ADD     #40,DA          ;INCREMENT TO THE NEXT CYL.
 2539  012366  022737  014540  001410          CMP     #14540,DA       ;ARE WE DONE WITH THIS ONE?
 2540  012374  001346                          BNE     NEXT            ;NO SO DO THE NEXT CYL.
 2541  012376  004237  013564          GOOD:   JSR     %2,TYPEIT       ;PUT OUT PACK GOOD MESSAGE.
 2542  012402  005015                          5015
 2543  012404  040520  045503  043440          .ASCIZ  /PACK GOOD./
 2544  012412  047517  027104     000
 2545          012420                          .EVEN
 2546  012420  000612                          BR      S11
 2547  012422  000137  001440          GD1:    JMP     @#START         ;RESTART
 2548                                          ;
 2549                                          ;
 2550                                  ;DISK I/O SUBROUTINE.
 2551                                          ;
 2552                                          ;
 2553                                  ;SET UP FOR A WRITE/FORMAT.
 2554  012426  013777  001412  166736  IO:     MOV     WC,@RKWC        ;SET UP THE WORD COUNT REG.
```

```
2555  012434  013777  001410  166734          MOV     DA,@RKDA        ;SET UP THE DISK ADDRESS.
2556  012442  053777  001352  166726          BIS     @#DSKTMP,@RKDA  ;SET THE UNIT NUMBER UP.
2557  012450  013777  001406  166716          MOV     BA,@RKBA        ;SET UP THE BUSS ADDRESS.
2558  012456  012777  000000  166704          MOV     #0,@RKCS        ;CLEAR THE CONTROL REG. FOR SET UP.
2559  012464  052777  006000  166676          BIS     #BIT10+BIT11,@RKCS ;SET FORMAT&INHIBT INC. BITS.
2560  012472  052777  000002  166670          BIS     #BIT1,@RKCS     ;SET UP WRITE FUN.
2561  012500  053777  001420  166662          BIS     EXTR,@RKCS      ;SET UP 12OR16 SECTOR PACK.
2562  012506  052777  000001  166654          BIS     #BIT0,@RKCS     ;GO DO THE WRITE FORMAT.
2563  012514  105777  166650          1$:     TSTB    @RKCS           ;IS WRITE FORMAT DONE?
2564  012520  100375                          BPL     1$              ;NO SO WAIT.
2565  012522  005777  166642                  TST     @RKCS           ;WAS THERE A ERROR?
2566  012526  100541                          BMI     WFERR           ;YES GO SERVICE IT.
2567  012530  012737  000000  001422          MOV     #0,ERRWF        ;CLEAR OUT THE ERROR COUNTER.
2568                                          ;
2569                                          ;SET UP FOR A READ/FORMAT
2570                                          ;
2571  012536  013777  001416  166626          MOV     RWC,@RKWC       ;SET UP WORD COUNT REG.
2572  012544  013777  001410  166624          MOV     DA,@RKDA        ;SET UP DISK ADDRESS.
2573  012552  053777  001352  166616          BIS     @#DSKTMP,@RKDA  ;SET THE UNIT NUMBER .
2574  012560  013777  001414  166606          MOV     RBA,@RKBA       ;SET UP THE BUSS ADDRESS.
2575  012566  012777  000000  166574          MOV     #0,@RKCS        ;CLEAR THE CONTROL REG.
2576  012574  052777  002000  166566          BIS     #BIT10,@RKCS    ;SET THE FORMAT BIT.
2577  012602  052777  000004  166560          BIS     #BIT2,@RKCS     ;SET UP READ FUN.
2578  012610  053777  001420  166552          BIS     EXTR,@RKCS      ;SET UP 12 OR 16 SECTOR PACK.
2579  012616  052777  000001  166544          BIS     #BIT0,@RKCS     ;GO DO THE READ FORMAT.
2580  012624  105777  166542          2$:     TSTB    @RKCS           ;IS THE READ FORMAT DONE?
2581  012630  100375                          BPL     2$              ;NO SO WAIT.
2582  012632  032777  040000  166530          BIT     #BIT14,@RKCS    ;WAS TRERE A ERROR?
2583  012640  001133                          BNE     RFERR           ;YES GO SERVICE IT.
2584  012642  012737  000000  001424          MOV     #0,ERRRF        ;CLEAR OUT THE ERROR COUNT.
2585                                          ;
2586                                          ;SET UP FOR A WRITE CHECK.
2587                                          ;
2588  012650  013777  001412  166514          MOV     WC,@RKWC        ;SET UP WORD COUNT REG.
2589  012656  013777  001410  166512          MOV     DA,@RKDA        ;SET UP DISK ADDRESS.
2590  012664  053777  001352  166504          BIS     @#DSKTMP,@RKDA  ;SET UP THE UNIT NUMBER
2591  012672  013777  001406  166474          MOV     BA,@RKBA        ;SET UP BUSS ADDRESS.
2592  012700  012777  000000  166462          MOV     #0,@RKCS        ;CLEAR THE CONTROL REG.
2593  012706  052777  004400  166454          BIS     #BIT11+BIT8,@RKCS ;SET INHIBIT INCR.&STOP ON SOFT ERROR BITS.
2594  012714  053777  001420  166446          BIS     EXTR,@RKCS      ;SET 12 OR 16 SECTOR PACK.
2595  012722  052777  000006  166440          BIS     #BIT1+BIT2,@RKCS ;SET UP WRITE CHECK FUN.
2596  012730  052777  000001  166432          BIS     #BIT0,@RKCS     ;GO DO THE WRITE CHECK.
2597                                          ;
2598                                          ;CHECK HEADERS READ BY THE READ/FORMAT.
2599                                          ;
2600  012736  013703  001416                  MOV     RWC,%3          ;PUT NUMBER OF WORDS TO
2601  012742  005403                          NEG     %3              ;CHECK IN REG 3.
2602  012744  063703  001414                  ADD     RBA,%3          ;SET REG 3 TO THE LAST WORD TO BE CHECKED.
2603  012750  013702  001414                  MOV     RBA,%2          ;SET REG 2 TO STARTING ADD. OF BUFF.
2604  012754  023722  001410          MORE:   CMP     DA,(2)+         ;CHECK THAT HEADER IS RIGHT.
2605  012760  001072                          BNE     RFCERR          ;THIS HEADER WAS WRONG GO SERVICE IT.
2606  012762  020302                          CMP     %3,%2           ;ARE WE DONE?
2607  012764  001373                          BNE     MORE            ;NO GO CHECK THE NEXT ONE.
2608  012766  012737  000000  001426          MOV     #0,ERRRFC       ;CLEAR OUT THE ERROR COUNT.
2609                                          ;
2610                                          ;LETS CHECK ON THE WRITE CHECK WE STARTED.
```

```
2611                                            ;
2612  012774  105777  166370          1$:       TSTB    @RKCS
2613  013000  100375                            BPL     1$              ;THE CONTROLER IS STILL BUSY.
2614  013002  005777  166362                    TST     @RKCS           ;WAS THERE A ERROR?
2615  013006  100407                            BMI     WCERRR          ;YES GO SERVICE IT.
2616  013010  012737  000000  001430            MOV     #0,ERRWCH       ;CLEAR OUT THE
2617  013016  012737  000000  001432            MOV     #0,ERRWCS       ;ERROR COUNTERS.
2618  013024  000201                            RTS     %1              ;RETURN TO THE MAIN LINE.
2619  013026  000137  013510          WCERRR:   JMP     WCERR
2620                                            ;
2621                                            ;ERRORS FOR WRITE FORMAT.
2622                                            ;
2623  013032  005237  001422          WFERR:    INC     ERRWF           ;ADD ONE TO THE ERROR COUNT.
2624  013036  022737  000004  001422            CMP     #4,ERRWF        ;HAS IT HAPPEND 4 TIMES ON THIS CYL.
2625  013044  001015                            BNE     RETRY           ;NO.
2626  013046  004237  013564          SYSER:    JSR     %2,TYPEIT       ;PUT OUT "SYSTEM ERROR" MESSAGE.
2627  013052  005015                            5015
2628  013054  054523  052123  046505            .ASCIZ  /SYSTEM ERROR./
2629  013062  042440  051122  051117
2630  013070  000056
2631                                            .EVEN
2632  013072  000000                            HALT                    ;LET THE TECH. BREATH.
2633  013074  000137  001440                    JMP     START           ;RESTART THE TEST.
2634  013100  012777  000000  166262  RETRY:    MOV     #0,@RKCS        ;CLEAR OUT THE CONTROL REG.
2635  013106  012777  000015  166254            MOV     #15,@RKCS       ;DO A DRIVE RESET.
2636  013114  032777  000100  166242  1$:       BIT     #BIT6,@RKDS     ;IS IT DONE.
2637  013122  001774                            BEQ     1$              ;NO SO WAIT.
2638  013124  000137  012426                    JMP     IO              ;TRY AGAIN.
2639                                            ;
2640                                            ;ERRORS FOR READ/FORMAT.
2641                                            ;
2642  013130  005237  001424          RFERR:    INC     ERRRF           ;ADDONE TO ERROR COUNT.
2643  013134  022737  000004  001424            CMP     #4,ERRRF        ;HAS IT HAPPEND 4 TIMES ON THIS CYL?
2644  013142  001356                            BNE     RETRY           ;NO DO IT AGAIN.
2645  013144  000740                            BR      SYSER           ;YES SO TELL HIM SO.
2646                                            ;
2647                                            ;READ/FORMAT ERRORS FOUND BY SOFTWARE CHECKS.
2648                                            ;
2649  013146  005237  001426          RFCERR:   INC     ERRRFC          ;ADD ONE TO ERROR COUNT.
2650  013152  105777  166212          1$:       TSTB    @RKCS           ;WAIT FOR THE WRITE CHECK.
2651  013156  100375                            BPL     1$
2652  013160  022737  000004  001426            CMP     #4,ERRRFC       ;IS IT 4?
2653  013166  001401                            BEQ     FAILED          ;PUT OUT FAILED MESSAGE.
2654  013170  000743                            BR      RETRY           ;NO SO GO TRY IT AGAIN.
2655  013172  042777  000037  166176  FAILED:   BIC     #37,@RKDA       ;PUT WHICH SECTORS HEADER
2656  013200  042702  177740                    BIC     #177740,%2
2657  013204  060277  166166                    ADD     %2,@RKDA        ;FAILED IN RKDA FOR THE MESSAGE.
2658  013210  004237  013564          FAIL:     JSR     %2,TYPEIT       ;TYPE OUT THE FAILED MESSAGE.
2659  013214  005015                            5015
2660  013216  105212                            LFLF
2661  013220  105212                            LFLF
2662  013222  040520  045503  043040            .ASCIZ  /PACK FAILED AT(IN OCTAL)/
2663  013230  044501  042514  020104
2664  013236  052101  044450  020116
2665  013244  041517  040524  024514
2666  013252     000
```

```
 2667            013254                          .EVEN
 2668   013254   017701   155116                 MOV     JRKDA,%1        ;GENERAT THE CYL,SECTOR,SURFACE
 2669   013260   010102                          MOV     %1,%2   ;MESSAGE FROM RKDA
 2670   013262   042702   177770                 BIC     #177770,%2
 2671   013266   062702   000260                 ADD     #260,%2
 2672   013272   110237   013445                 MOVB    %2,SEC+1
 2673   013276   004337   013476                 JSR     %3,SHF3
 2674   013302   042702   177776                 BIC     #177776,%2
 2675   013306   062702   000260                 ADD     #260,%2
```

```
 2676  013312  110237  013444          MOVB    %2,SEC
 2677  013316  004337  013502          JSR     %3,SHF1
 2678  013322  042702  177776          BIC     #177776,%2
 2679  013326  062702  000260          ADD     #260,%2
 2680  013332  110237  013456          MOVB    %2,SUR
 2681  013336  004337  013502          JSR     %3,SHF1
 2682  013342  042702  177770          BIC     #177770,%2
 2683  013346  062702  000260          ADD     #260,%2
 2684  013352  110237  013434          MOVB    %2,CYL+2
 2685  013356  004337  013476          JSR     %3,SHF3
 2686  013362  042702  177770          BIC     #177770,%2
 2687  013366  062702  000260          ADD     #260,%2
 2688  013372  110237  013433          MOVB    %2,CYL+1
 2689  013376  004337  013476          JSR     %3,SHF3
 2690  013402  042702  177774          BIC     #177774,%2
 2691  013406  062702  000260          ADD     #260,%2
 2692  013412  110237  013432          MOVB    %2,CYL
 2693  013416  004237  013564          JSR     %2,TYPEIT       ;TYPE OUT THE GENERATED MESSAGE.
 2694  013422  005015                  5015
```

```
 2695  013424  054503  027114  020040          .ASCII  /CYL.  /
 2696  013432  030060  020060          CYL:    .ASCII  /000 /
 2697  013436  042523  027103  020040          .ASCII  /SEC.  /
 2698  013444  030060     040          SEC:    .ASCII  /00 /
 2699  013447     123  051125  027106          .ASCII  /SURF.  /
 2700  013454  020040
 2701  013456  020060                  SUR:    .ASCII  /0 /
 2702  013460  005015                          5015
 2703  013462  105212                          LFLF
 2704  013464  105212                          LFLF
 2705  013466  000040                          .ASCIZ  / /
 2706                                           .EVEN
 2707  013470  000000                  STOP:   HALT                    ;LET OPER DO HIS THING.
 2708  013472  000137  001440                  JMP     START           ;RESTART THE TEST.
 2709                                  ;
 2710                                  ;SHIFT SUBROUTINES.
 2711                                  ;
 2712  013476  006201                  SHF3:   ASR     %1              ;HERE FOR A SHIFT OF 3.
 2713  013500  006201                  SHF2:   ASR     %1              ;HERE FOR A SHIFT OF 2.
 2714  013502  006201                  SHF1:   ASR     %1              ;HERE FOR A SHIFT OF 1.
 2715  013504  010102                          MOV     %1,%2           ;PUT RESULTES IN THE WORKING REG.
 2716  013506  000203                          RTS     %3
 2717                                  ;
 2718                                  ;ERRORS FOR WRITE CHECK.
 2719                                  ;
 2720  013510  032777  040000  165652  WCERR:  BIT     #BIT14,@RKCS    ;WAS IT A HARD ERROR.
 2721  013516  001010                          BNE     WCHERR          ;YES GO PROSSES IT.
 2722  013520  005237  001432                  INC     ERRWCS          ;ADD 1 TO THE SOFT ERROR COUNT.
 2723  013524  022737  000004  001432          CMP     #4,ERRWCS       ;HAS THERE BEEN 4 OF THEM?
 2724  013532  001626                          BEQ     FAIL            ;YES PUT OUT FAILED MESSAGE.
 2725  013534  000137  012426                  JMP     IO              ;NO SO TRY AGAIN.
 2726  013540  005237  001430          WCHERR: INC     ERRWCH          ;ADD 1 TO THE HARD ERROR COUNT.
 2727  013544  022737  000004  001430          CMP     #4,ERRWCH       ;HAS THERE BEEN 4 OF THEM?
 2728  013552  001402                          BEQ     SYSERR          ;YES PUT OUT SYSTEM ERROR MESSAGE.
 2729  013554  000137  013100                  JMP     RETRY           ;NO SO TRY AGAIN.
 2730  013560  000137  013046          SYSERR: JMP     SYSER
 2731                                  ;
 2732                                  ;TYPE OUT SUBROUTINE.
 2733                                  ;
 2734  013564  105777  165360          TYPEIT: TSTB    @$TPS           ;IS TTY BUSY?
 2735  013570  100375                          BPL     TYPEIT          ;YES SO WAIT.
 2736  013572  105712                          TSTB    (2)             ;IS THIS A 00 BYTE?
 2737  013574  001403                          BEQ     END             ;YES THEN WE ARE DONE.
 2738  013576  112277  165350                  MOVB    (2)+,@$TPB      ;NO SO PRINT IT.
 2739  013602  000770                          BR      TYPEIT          ;GO BACK AND WAIT.
 2740  013604  005202                  END:    INC     %2              ;MOVE OVER 00 BYTE.
 2741  013606  032702  000001                  BIT     #1,%2           ;IS THE RETURN REG. ODD?
 2742  013612  001401                          BEQ     .+4             ;NO SO RETURN.
 2743  013614  005202                          INC     %2              ;YES MAKE IT EVEN.
 2744  013616  000202                          RTS     %2              ;GO BACK.
 2745                                  ;
 2746                                  ;BUFFERS.
 2747                                  ;
 2748  013620  000000                  BUFF:   .WORD   0
 2749  013622  000030                  RBUFF:  .BLKW   30
 2750
```

```
                                                BAD.ON:
013702                                                  TYPE    ,65$            ;;TYPE ASCIZ STRING
013702  104401  013710                                  BR      64$             ;;GET OVER THE ASCIZ
013706  000401                                  ;;65$:  .ASCIZ  /?/
                                                64$:
013712                                                  .SBTTL  RK05 CONTROL PANEL TEST
                                                SECT.0:
013712                                                  TYPE    ,65$            ;;TYPE ASCIZ STRING
013712  104401  013720                                  BR      64$             ;;GET OVER THE ASCIZ
013716  000424                                  ;;65$:  .ASCIZ  <15><12>/RK05 CONTROL PANEL TEST, WHICH DRIVE?/
                                                64$:
013770                                                  RDCHR
013770  104405                                          MOV     (SP),R0
013772  011600                                          TYPOS
013774  104403                                          .BYTE   1
013776     001                                          .BYTE   0
013777     000                                          SUB     #60,R0
014000  162700  000060                                  BMI     BAD.ON
014004  100736                                          CMP     #10,R0
014006  022700  000010                                  BLE     BAD.ON
014012  003733                                          MOVB    R0,@#DRIVE
014014  110037  001314                                  SWAB    R0
014020  000300                                          ROL     R0
014022  006100                                          ROL     R0
014024  006100                                          ROL     R0
014026  006100                                          ROL     R0
014030  006100                                          MOV     R0,@#DSKTMP
014032  006100                                          TYPE    ,67$            ;;TYPE ASCIZ STRING
014034  010037  001352                                  BR      66$             ;;GET OVER THE ASCIZ
014040  104401  014046                          ;;67$:  .ASCIZ  <15><12>/MOUNT PACK ON DRIVE#/
014044  000414                                  66$:
                                                        MOV     DRIVE,-(SP)     ;;SAVE DRIVE FOR TYPEOUT
014076                                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
014076  013746  001314                                  .BYTE   1               ;;TYPE 1 DIGIT(S)
014102  104403                                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
014104     001                                          TYPE    ,69$            ;;TYPE ASCIZ STRING
014105     000                                          BR      68$             ;;GET OVER THE ASCIZ
014106  104401  014114                          ;;69$:  .ASCIZ  <15><12>/PLACE DRIVE IN RUN ;SHOULD SEE THE RUN.
014112  000425                                  68$:
014166                                                  TYPE    ,71$            ;;TYPE ASCIZ STRING
014166  104401  014174                                  BR      70$             ;;GET OVER THE ASCIZ
014172  000423                                  ;;71$:  .ASCIZ  <15><12>/POWER, AND ON CYLINDER LAMPS LIGHT.
                                                70$:
014242                                                  TYPE    ,73$            ;;TYPE ASCIZ STRING
014242  104401  014250                                  BR      72$             ;;GET OVER THE ASCIZ
014246  000425                                  ;;73$:  .ASCIZ  <15><12>/MAKE DRIVE WRITE ENABLE PRESS CONTINUE
                                                72$:
014322                                                  HALT
014322  000000                                          JSR     PC,WRROCK       ;GO WRITE 0'S, RD 0'S, CHECK
014324  004737  016102                                  TYPE    ,75$            ;;TYPE ASCIZ STRING
014330  104401  014336
```

```
 2807  014334  000430                          BR      74$             ;;GET OVER THE ASCIZ
 2808                                  ;;75$:  .ASCIZ  <15><12><15><12>/WRITE PROTECT THE DRIVE THEN PRESS CONTINUE/
 2809  014416                          74$:
 2810  014416  000000                          HALT
 2811  014420  004737  016312                  JSR     PC,WRPRO        ;GO TRY OVERWRITE
 2812  014424  104401  014432                  TYPE    ,77$            ;;TYPE ASCIZ STRING
 2813  014430  000426                          BR      76$             ;;GET OVER THE ASCIZ
 2814                                  ;;77$:  .ASCIZ  <15><12><15><12>/CLEAR WRITE PROTECT THEN PRESS CONTINUE/
 2815  014506                          76$:
 2816  014506  000000                          HALT
 2817  014510  004737  016102                  JSR     PC,WRRDCK       ;GO WRITE 0'S, RD 0'S, CHECK
 2818  014514  013777  001352  164654          MOV     @#DSKTMP,@RKDA  ;SET UP DRIVE#
 2819  014522  012777  000017  164640          MOV     #17,@RKCS       ;FUNCTION WRITE PROTECT
 2820  014530  004737  005554                  JSR     PC,$MTME        ;GO WAIST TIME FOR RK11-C
 2821  014534  105777  164630          1$:     TSTB    @RKCS           ;IS CONTROL READY SET
 2822  014540  100375                          BPL     1$              ;IF NO, BRANCH
 2823  014542  004737  016312                  JSR     PC,WRPRO        ;GO TRY OVERWRITE OF IERO'S
 2824  014546                          PRTTWO:
 2825  014546  104401  014554                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 2826  014552  000431                          BR      64$             ;;GET OVER THE ASCIZ
 2827                                  ;;65$:  .ASCIZ  <15><12><15><12>/CAUTION! TRY TO OPEN THE DOOR, DO NOT FORCE:/
 2828  014636                          64$:
 2829  014636  104401  014644                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 2830  014642  000414                          BR      66$             ;;GET OVER THE ASCIZ
 2831                                  ;;67$:  .ASCIZ  <15><12>/DOOR SHOULD NOT OPEN!/
 2832  014674                          66$:
 2833  014674  104401  014702                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 2834  014700  000420                          BR      68$             ;;GET OVER THE ASCIZ
 2835                                  ;;69$:  .ASCIZ  <15><12>/PRESS CONTINUE WHEN FINISHED/
 2836  014742                          68$:
 2837  014742  000000                          HALT
 2838  014744  104401  014752                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 2839  014750  000426                          BR      70$             ;;GET OVER THE ASCIZ
 2840                                  ;;71$:  .ASCIZ  <15><12><15><12>/PUT DRIVE IN LOAD, WAIT FOR LOAD LIGHT
 2841  015026                          70$:
 2842  015026  104401  015034                  TYPE    ,73$            ;;TYPE ASCIZ STRING
 2843  015032  000420                          BR      72$             ;;GET OVER THE ASCIZ
 2844                                  ;;73$:  .ASCIZ  <15><12>/PRESS CONTINUE WHEN FINISHED/
 2845  015074                          72$:
 2846  015074  000000                          HALT
 2847  015076  013777  001352  164272          MOV     @#DSKTMP,@RKDA  ;SET UP DISK ADDRESS
 2848  015104  012777  000015  164256          MOV     #15,@RKCS       ;ISSUE A DRIVE RESET
 2849  015112  004737  005554                  JSR     PC,$MTME        ;KILL TIME FOR RK11-C
 2850  015116  105777  164246          1$:     TSTB    @RKCS           ;CONTROL READY SET?
 2851  015122  100375                          BPL     1$              ;IF NO, BRANCH
 2852  015124  032777  100000  164234          BIT     #BIT15,@RKER    ;DRE SET
 2853  015132  001023                          BNE     2$              ;IF YES BRANCH
 2854  015134  104401  015142                  TYPE    ,75$            ;;TYPE ASCIZ STRING
 2855  015140  000420                          BR      74$             ;;GET OVER THE ASCIZ
 2856                                  ;;75$:  .ASCIZ  <15><12>/DRE=BIT15 OF RKER DID NOT SET
 2857  015202                          74$:
 2858  015202  032777  140000  164160  2$:     BIT     #140000,@RKCS   ;DID HARD ERROR OR ERROR SET
 2859  015210  001015                          BNE     3$              ;BRANCH IF YES
 2860  015212  104401  015220                  TYPE    ,77$            ;;TYPE ASCIZ STRING
 2861  015216  000412                          BR      76$             ;;GET OVER THE ASCIZ
 2862                                  ;;77$:  .ASCIZ  <15><12>/ERROR DID NOT SET/
```

```
 2963  015244                                    76$:
 2964  015244  012777  000001  164116  3$:       MOV     #1,@RKCS          ;ISSUE A CONTROL RESET
 2965  015252  004737  005554                    JSR     PC,SMTME          ;WAIST TIME
 2966  015256  105777  164106            4$:     TSTB    @RKCS             ;CONTROL READY SET
 2967  015262  100375                            BPL     4$                ;IF NO, BRANCH
 2968  015264  032777  100000  164074            BIT     #BIT15,@RKER      ;'DRE' CLEAR
 2969  015272  001425                            BEQ     5$                ;IF YES BRANCH
 2970  015274  104401  015302                    TYPE    ,79$              ;;TYPE ASCIZ STRING
 2971  015300  000422                            BR      78$               ;;GET OVER THE ASCIZ
 2972                                    ;;79$:   .ASCIZ  <15><12>/CONTROL RESET DID NOT CLEAR 'DRE'/
                                         78$:
 2973  015346                                    
 2974  015346  032777  140000  164014  5$:       BIT     #140000,@RKCS     ;ERROR BITS CLEAR
 2975  015354  001431                            BEQ     X                 ;IF YES BRANCH
 2976  015356  104401  015364                    TYPE    ,81$              ;;TYPE ASCIZ STRING
 2977  015362  000426                            BR      80$               ;;GET OVER THE ASCIZ
 2978                                    ;;81$:   .ASCIZ  <15><12>/CONTROL RESET DID NOT CLEAR 'ERROR', RKCS/
 2979  015440                           80$:
 2980  015440                           X:
 2981  015440  104401  015446                    TYPE    ,65$              ;;TYPE ASCIZ STRING
 2982  015444  000422                            BR      64$               ;;GET OVER THE ASCIZ
 2983                                    ;;65$:   .ASCIZ  <15><12><15><12>/OPEN THE DOOR, PUT DRIVE IN RUN/
 2984  015512                           64$:
 2985  015512  104401  015520                    TYPE    ,67$              ;;TYPE ASCIZ STRING
 2986  015516  000425                            BR      66$               ;;GET OVER THE ASCIZ
 2987                                    ;;67$:   .ASCIZ  <15><12>/CAUTION! IF RUN LIGHT ON ERROR! DEPRESS/
 2988  015572                           66$:
 2989  015572  104401  015600                    TYPE    ,69$              ;;TYPE ASCIZ STRING
 2990  015576  000426                            BR      68$               ;;GET OVER THE ASCIZ
 2991                                    ;;69$:   .ASCIZ  <15><12>/LOAD IMMEDIATELY, CONTINUE WHEN FINISHED/
 2992  015654                           68$:
 2993  015654  000000                   XX:      HALT
 2994  015656  104401  015664                    TYPE    ,65$              ;;TYPE ASCIZ STRING
 2995  015662  000422                            BR      64$               ;;GET OVER THE ASCIZ
 2996                                    ;;65$:   .ASCIZ  <15><12><15><12>/REMOVE THE PACK, CLOSE THE DOOR/
 2997  015730                           64$:
 2998  015730  104401  015736                    TYPE    ,67$              ;;TYPE ASCIZ STRING
 2999  015734  000423                            BR      66$               ;;GET OVER THE ASCIZ
 3000                                    ;;67$:   .ASCIZ  <15><12>/PUT DRIVE IN RUN, DRIVE SHOULD NOT/
 3001  016004                           66$:
 3002  016004  104401  016012                    TYPE    ,69$              ;;TYPE ASCIZ STRING
 3003  016010  000423                            BR      68$               ;;GET OVER THE ASCIZ
 3004                                    ;;69$:   .ASCIZ  <15><12>/RUN...INTERCLOCKS HAVE BEEN CHECKED/
 3005  016060                           68$:
 3006  016060  104401  016066                    TYPE    ,71$              ;;TYPE ASCIZ STRING
 3007  016064  000404                            BR      70$               ;;GET OVER THE ASCIZ
 3008                                    ;;71$:   .ASCIZ  <15><12>/DONE!/
 3009  016076                           70$:
 3010  016076  000137  001440                    JMP     @#START
 3011  016102  005037  001362          WRRDCK:  CLR     @#PATTRN          ;MAKE A PATTERN OF ZERO'S
 3012  016106  013777  001352  163262            MOV     @#DSKTMP,@RKDA    ;SET UP DRIVE ADDRESS
 3013  016114  012777  000001  163246            MOV     #1,@RKCS          ;ISSUE A CONTROL RESET
 3014  016122  004737  005554                    JSR     PC,SMTME
 3015  016126  105777  163236            5$:     TSTB    @RKCS             ;CONTROL READY SET
 3016  016132  100375                            BPL     5$                ;IF NO BRANCH
 3017  016134  013777  001352  163234            MOV     @#DSKTMP,@RKDA    ;SET UP DRIVE ADDRESS
 3018  016142  012777  001362  163224            MOV     #PATTRN,@RKBA     ;GET BUSS ADDRESS
```

```
2919  016150  013777  001344  163214          MOV     @#SECCNT,@RKWC  ;WORD COUNT 1 SECTOR
2920  016156  013777  001354  163204          MOV     @#WRITCS,@RKCS  ;IBA + WRITE + GO
2921  016164  004737  005554                  JSR     PC,SMTME        ;KILL TIME FOR RK11-C
2922  016170  105777  163174          1$:     TSTB    @RKCS           ;CONTROL READY SET
2923  016174  100375                          BPL     1$              ;IF NO, BRANCH
2924  016176  013777  001352  163172          MOV     @#DSKTMP,@RKDA  ;SET UP RK REGISTERS
2925  016204  012777  007316  163162          MOV     #RDBUFF,@RKBA   ;TO READ ONE SECTOR
2926  016212  013777  001344  163152          MOV     @#SECCNT,@RKWC  ;TO THE READ BUFFER
2927  016220  013777  001356  163142          MOV     @#READCS,@RKCS
2928  016226  004737  005554                  JSR     PC,SMTME        ;KILL TIME FOR RK11-C
2929  016232  105777  163132          2$:     TSTB    @RKCS           ;CONTROL READY SET
2930  016236  100375                          BPL     2$              ;IF NO, BRANCH
2931  016240  012704  007316                  MOV     #RDBUFF,R4      ;GET BUFFER TO R4
2932  016244  005005                          CLR     R5              ;SET UP TO COMPARE
2933  016246  020524          3$:     CMP     R5,(R4)+        ;FOR ZERO'S
2934  016250  001414                          BEQ     4$              ;IF OK, BRANCH
2935  016252  104401  016260                  TYPE    ,65$            ;;TYPE ASCIZ STRING
2936  016256  000410                          BR      64$             ;;GET OVER THE ASCIZ
2937                          ;;65$:  .ASCIZ  <15><12>/WRITE FAILED/
2938  016300                  64$:
2939  016300  000700                          BR      WRRDCK          ;GO BACK TRY AGAIN
2940  016302  022704  010316          4$:     CMP     #MANSEL,R4      ;DONE ALL CHECKS
2941  016306  001357                          BNE     3$              ;IF NO, BRANCH
2942  016310  000207                          RTS     PC              ;IF YES, RETURN
2943  016312  032777  000040  163044  WRPRO:  BIT     #BIT5,@RKDS     ;BIT 5 ON
2944  016320  001021                          BNE     1$              ;IF YES, BRANCH
2945  016322  104401  016330                  TYPE    ,65$            ;;TYPE ASCIZ STRING
2946  016326  000416                          BR      64$             ;;GET OVER THE ASCIZ
2947                          ;;65$:  .ASCIZ  <15><12>/WPS=BIT5 OF RKDS NOT SET/
2948  016364                  64$:
2949  016364  012737  177777  001362  1$:     MOV     #177777,@#PATTRN ;GO LOAD ALL
2950  016372  013777  001352  162776          MOV     @#DSKTMP,@RKDA  ;RK REGISTERS
2951  016400  012777  001362  162766          MOV     #PATTRN,@RKBA   ;TO WRITE
2952  016406  013777  001344  162756          MOV     @#SECCNT,@RKWC  ;ALL ONES (WITH WRITE LOCK)
2953  016414  013777  001354  162746          MOV     @#WRITCS,@RKCS  ;OVER THE ZERO'S
2954  016422  004737  005554                  JSR     PC,SMTME        ;KILL TIME FOR RK11-C
2955  016426  105777  162736          2$:     TSTB    @RKCS           ;CONTROL READY SET?
2956  016432  100375                          BPL     2$              ;IF NO, BRANCH
2957  016434  032777  [illegible]  162724      BIT     #BIT13,@RKER    ;WLO BIT SET
2958  016442  001032                          BNE     3$              ;IF YES BRANCH
2959  016444  104401  016452                  TYPE    ,67$            ;;TYPE ASCIZ STRING
2960  016450  000427                          BR      66$             ;;GET OVER THE ASCIZ
2961                          ;;67$:  .ASCIZ  <15><12>/EXPECTED WLO=BIT13 OF RKER BUT DID NOT SET.
2962  016530                  66$:
2963  016530  012777  000001  162632  3$:     MOV     #1,@RKCS        ;DO A CONTROL RESET
2964  016536  004737  005554                  JSR     PC,SMTME        ;KILL TIME FOR RK11-C
2965  016542  105777  162622          4$:     TSTB    @RKCS           ;CONTROL READY SET?
2966  016546  100375                          BPL     4$              ;IF NO BRANCH
2967  016550  032777  020000  162610          BIT     #BIT13,@RKER    ;WLO BIT CLEAR
2968  016556  001431                          BEQ     RDCHK0          ;IF YES, BRANCH
2969  016560  104401  016566                  TYPE    ,69$            ;;TYPE ASCIZ STRING
2970  016564  000426                          BR      68$             ;;GET OVER THE ASCIZ
2971                          ;;69$:  .ASCIZ  <15><12>/CONTROL RESET DID NOT CLEAR 'WLO' OF RKER
2972  016642                  68$:
2973  016642  013777  001352  162526  RDCHK0: MOV     @#DSKTMP,@RKDA  ;SET UP RK REGISTERS
2974  016650  012777  007316  162516          MOV     #RDBUFF,@RKBA   ;TO READ SECTOR 0,
```

```
 2975  016656  013777  001344  162506          MOV     @#SECCNT,@RKWC  ;CYLINDER 0, HEAD 0
 2976  016664  013777  001356  162476          MOV     @#READCS,@RKCS  ;TO ENSURE NO WRITE TOOK PLACE
 2977  016672  004737  005554                  JSR     PC,SMTME        ;KILL TIME
 2978  016676  105777  162466          3$:     TSTB    @RKCS
 2979  016702  100375                          BPL     3$
 2980  016704  012703  000005                  MOV     #5,R3
 2981  016710  012704  007316                  MOV     #RDBUFF,R4      ;CHECK TO INSURE NO WRITE
 2982  016714  005005                          CLR     R5              ;TOOK PLACE
 2983  016716  020524          1$:     CMP     R5,(R4)+        ;WITH WRITE LOCK
 2984  016720  001474                          BEQ     2$
 2985  016722  005303                          DEC     R3              ;DEC THE ERROR COUNT
 2986  016724  001475                          BEQ     4$              ;IF ZERO BRANCH
 2987  016726  104401  016734                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 2988  016732  000422                          BR      64$             ;;GET OVER THE ASCIZ
 2989                          ;;65$:  .ASCIZ  <15><12>/WRITE OCCURRED WITH WRITE PROTECT/
 2990  017000                  64$:
 2991  017000  005744                          TST     -(R4)
 2992  017002  104401  017010                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 2993  017006  000410                          BR      66$             ;;GET OVER THE ASCIZ
 2994                          ;;67$:  .ASCIZ  <15><12>/BUFFER ADDR=/
 2995  017030                  66$:
 2996  017030  010446                          MOV     R4,-(SP)
 2997  017032  104403                          TYPOS
 2998  017034     006                          .BYTE   6
 2999  017035     001                          .BYTE   1
 3000  017036  104401  017044                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 3001  017042  000406                          BR      68$             ;;GET OVER THE ASCIZ
 3002                          ;;69$:  .ASCIZ  /    EXPCTD=/
 3003  017060                  68$:
 3004  017060  010546                          MOV     R5,-(SP)
 3005  017062  104404                          TYPON
 3006  017064  104401  017072                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 3007  017070  000406                          BR      70$             ;;GET OVER THE ASCIZ
 3008                          ;;71$:  .ASCIZ  /    RECVD=/
 3009  017106                  70$:
 3010  017106  012446                          MOV     (R4)+,-(SP)
 3011  017110  104404                          TYPON
 3012  017112  022704  010316          2$:     CMP     #MANSEL,R4      ;FINISHED ALL CHECKS
 3013  017116  001277                          BNE     1$              ;IF NO, BRANCH
 3014  017120  000207          4$:     RTS     PC              ;RETURN
 3015
 3016
 3017
 3018
```

```
                                        ;THE FOLLOWING REVISION WAS MADE BY JIM KAPADIA

                                                .SBTTL  CONTROL PANEL TEST # 2

                                        ;THIS IS THE ENTRY POINT INTO CONTROL PANEL TEST #2.  ALL
                                        ;THE DRIVES THAT ARE PRESENT AND IN 'RDY' CONDITION ARE
                                        ;REPORTED (ON LINE).

017122  000240                  SECT.4: NOP
017124  012777  000001  162236          MOV     #1,@RKCS
017132  105777  162232          3$:     TSTB    @RKCS
017136  100375                          BPL     3$
017140  012700  020436                  MOV     #DRIV0,R0
017144  005001                          CLR     R1
017146  005002                          CLR     R2

017150  010210                  1$:     MOV     R2,(R0)         ;SET UP ADDRESS TABLE
017152  010277  162220                  MOV     R2,@RKDA        ;ADDRESS THE DRIVE
017156  105777  162202                  TSTB    @RKDS           ;IS IT PRESENT?
017162  100021                          BPL     2$              ;NO

017164  104401  020272                  TYPE    ,EM1            ;TYPE 'DRIVE'
017170  010146                          MOV     R1,-(SP)        ;TYPE OUT DRIVE #
017172  104402                          TYPOC
017174  104401  020303                  TYPE    ,EM2            ;TYPE 'ON LINE'
017200  052710  000300                  BIS     #BIT6+BIT7,(R0) ;SET BITS INDICATING THIS
                                                                ;DRIVE PRESENT

017204  012777  000015  162156          MOV     #15,@RKCS       ;ISSUE A DRIVE RESET
017212  004737  005554                  JSR     PC,$MTME        ;ALLOW SOME TIME
017216  032777  000100  162140  4$:     BIT     #RWS,@RKDS      ;WAIT FOR RWS RDY
017224  001774                          BEQ     4$

017226  005720                  2$:     TST     (R0)+
017230  005201                          INC     R1
017232  062702  020000                  ADD     #20000,R2       ;NXT DRIVE
017236  001344                          BNE     1$              ;ALL DONE?

017240  104401  001161                  TYPE    ,$CRLF

                                        ;THIS CODE CHECKS THE CONDITION OF 'DRY' BIT IN RKDS FOR EVERY
                                        ;DRIVE.  IF 'DRY' IS SET DRIVE IS SAID TO BE 'ON LINE', OTHERWISE IT
                                        ;IS OFFLINE.  IF THE 'DRY' BIT HAS CHANGED FROM LAST TIME, THEN
                                        ;IT IS REPORTED.  IF THERE IS NO CHANGE NOTHING IS REPORTED.

017244  012700  020436          BEGCT:  MOV     #DRIV0,R0       ;INITIALIZE POINTERS
017250  005001                          CLR     R1

017252  011077  162120          BEGCT1: MOV     (R0),@RKDA      ;ADDRESS A DRIVE
017256  042777  017777  162112          BIC     #17777,@RKDA    ;MASK OUT NON DR# BI S
017264  105777  162074                  TSTB    @RKDS           ;IS THIS DRIVE ON LINE?
017270  100044                          BPL     1$              ;NO
                                                                ;YES
017272  105710                          TSTB    (R0)            ;WAS IT 'ON LINE' LAST TIME?
017274  100454                          BMI     NXT1            ;YES, NO MESSAGE TO REPORT
017276  052710  000200                  BIS     #BIT7,(R0)      ;IT CHANGED FROM OFF LINE TO ON
```

```
3075  017302  104401  020272                  TYPE    ,EM1            ;LINE, REPORT MESSAGE
3076  017306  010146                          MOV     R1,-(SP)
3077  017310  104402                          TYPOC
3078  017312  104401  020303                  TYPE    ,EM2            ;TYPE 'ON LINE'
3079  017316  032777  000040  162040          BIT     #WPS,@RKDS      ;WRITE ENABLED?
3080  017324  001417                          BEQ     2$              ;YES, OK
3081  017326  104401  017334                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3082  017332  000414                          BR      64$             ;;GET OVER THE ASCIZ
3083                                  ;;65$:  .ASCIZ  <15><12>/EROR,NOT WRT ENABLED/
3084  017364                          64$:
3085  017364  012777  000017  161776  2$:     MOV     #17,@RKCS       ;WRITE PROT THE DISK
3086  017372  105777  161772          3$:     TSTB    @RKCS
3087  017376  100375                          BPL     3$
3088  017400  000412                          BR      NXT1
3089
3090  017402  105710                  1$:     TSTB    (R0)            ;WAS THIS DRIVE OFF LINE LAST
3091  017404  100010                          BPL     NXT1            ;TIME?  BRNCH IF YES
3092  017406  104401  020272                  TYPE    ,EM1            ;IF NOT, REPORT THE CHANGE
3093  017412  010146                          MOV     R1,-(SP)        ;TYPE DRIVE #
3094  017414  104402                          TYPOC
3095  017416  104401  020315                  TYPE    ,EM3            ;TYPE 'OFF LINE'
3096  017422  042710  000200                  BIC     #BIT7,(R0)      ;CLEAR BIT TO INDICATE THIS
3097                                                                  ;DRIVE 'OFF LINE'
3098
3099                                  ;THIS CODE CHECKS 'WPS' BIT FOR EVERY DRIVE THAT IS IN 'DRY'
3100                                  ;CONDITION (ON LINE).  IT REPORTS ANY CHANGE IN THE CONDITION OF
3101                                  ;THE 'WPS' BIT.  IF THERE WAS NO CHANGE FROM LAST TIME NOTHING
3102                                  ;IS REPORTED.  AT THE TIME OF ENTRY R0 POINTS TO DRIVE FLAG.
3103
3104  017426  105777  161732          NXT1:   TSTB    @RKDS           ;IS THIS DRIVE PRESENT?
3105                                                                  ;RKDA CONTAINS THE DRV #
3106  017432  100033                          BPL     NXT2            ;NO, SKIP CHECKING
3107
3108  017434  032777  000040  161722          BIT     #WPS,@RKDS      ;WPS BIT SET?
3109  017442  001014                          BNE     1$              ;YES
3110                                                                  ;WPS BIT CLEAR
3111  017444  032710  000004                  BIT     #BIT2,(R0)      ;WAS IT CLR LAST TIME ALSO?
3112  017450  001424                          BEQ     NXT2            ;YES, NOTHING TO REPORT.
3113                                                                  ;WPS CHANGED FROM 'SET'
3114  017452  104401  020272                  TYPE    ,EM1            ;TO 'CLR', REPORT IT
3115  017456  010146                          MOV     R1,-(SP)        ;TYPE DRIVE #
3116  017460  104402                          TYPOC
3117  017462  042710  000004                  BIC     #BIT2,(R0)      ;INDICATE THAT 'WPS' IS CLEAR
3118  017466  104401  020343                  TYPE    ,EM5            ;TYPE 'WPS CLEAR'
3119  017472  000413                          BR      NXT2
3120                                                                  ;WPS BIT IS SET
3121  017474  032710  000004          1$:     BIT     #BIT2,(R0)      ;WAS IT SET LAST TIME ALSO?
3122  017500  001010                          BNE     NXT2            ;YES, NOTHING TO REPORT.
3123                                                                  ;WPS CHANGED, FROM 'CLR' TO
3124                                                                  ;'SET', REPORT THIS CHANGE
3125  017502  104401  020272                  TYPE    ,EM1            ;TYPE 'DRIVE'
3126  017506  010146                          MOV     R1,-(SP)        ;TYPE DRIVE #
3127  017510  104402                          TYPOC
3128  017512  104401  020330                  TYPE    ,EM4            ;TYPE 'WPS SET'
3129  017516  052710  000004                  BIS     #BIT2,(R0)      ;SET FLAG BIT INDICATING WPS SET
3130
```

```
                                        ;THIS CODE PERFORMS A SEEK FUNCTION ON A DRIVE AND CHECKS IF
                                        ;THE 'DPL' BIT SET AS A RESULT. (IF THE POWER WAS CUT OFF
                                        ;FROM THE DRIVE).  NOTE THAT ONLY THOSE DRIVES ARE
                                        ;CHECKED WHICH WERE FOUND TO BE PRESENT AT BEGINNING (WHEN
                                        ;THIS TEST WAS ENTERED).  SEEK IS DONE TO CYLINDER 1.
                                        ;AT THE TIME OF ENTRY R0 POINTS TO THE DRIVE FLAG.

017522  032710  000100          NXT2:   BIT     #BIT6,(R0)      ;WAS THIS DRIVE PRESENT AT BEGNG
017526  001403                          BEQ     4$              ;NO
017530  105777  161630                  TSTB    @RKDS           ;IS IT PRESENT NOW?
017534  100402                          BMI     3$              ;YES
017536  000137  020174          4$:     JMP     DN1DRV          ;IF NOT SKIP THIS CHECK

017542  052777  000040  161626  3$:     BIS     #40,@RKDA       ;RKDA ALREADY HAS THE DRV #
                                                                ;SET CYL 1 ADDRESS
017550  012777  000011  161612          MOV     #11,@RKCS       ;SEEK, GO

017556  105777  161606          1$:     TSTB    @RKCS           ;WAIT FOR CONTROL RDY?
017562  100375                          BPL     1$              ;SOMETHING WRONG IF CNTRL RDY
                                                                ;DOES NOT COME BACK
017564  032777  010000  161572          BIT     #DPL,@RKDS      ;DPL BIT SET?
017572  001414                          BEQ     2$              ;NO
                                                                ;YES, DPL SET
017574  032710  000001                  BIT     #BIT0,(R0)      ;WAS 'DPL' SET LAST TIME ALSO?
017600  001167                          BNE     CLRDPL          ;YES, NOTHING TO REPORT.
                                                                ;DPL CHANGED, GOT SET THIS
017602  104401  020272                  TYPE    ,EM1            ;TIME, REPORT IT
017606  010146                          MOV     R1,-(SP)
017610  104402                          TYPOC
017612  104401  020361                  TYPE    ,EM6            ;TYPE 'POWER LO'
017616  052710  000001                  BIS     #BIT0,(R0)      ;SET FLAG BIT INDICATING THAT
                                                                ;DPL SET THIS TIME
017622  000556                          BR      CLRDPL

                                                                ;'DPL' BIT IS CLEAR
017624  032710  000001          2$:     BIT     #BIT0,(R0)      ;WAS 'DPL' CLEAR LAST TIME ALSO?
017630  001410                          BEQ     WATSK           ;YES, NOTHING TO REPORT
                                                                ;REPORT THAT 'DPL' BIT CHANGED,
017632  104401  020272                  TYPE    ,EM1            ;FROM SET TO CLEAR
017636  010146                          MOV     R1,-(SP)
017640  104402                          TYPOC
017642  104401  020402                  TYPE    ,EM7            ;TYPE 'POWER UP'
017646  042710  000001                  BIC     #BIT0,(R0)      ;SET FLAG BIT INDICATING THAT DPL
                                                                ;IS CLEAR THIS TIME


                                ;THIS CODE WAITS FOR THE SEEK (DONE ABOVE) TO FINISH.  WAITING
                                ;TIME IS APPROX. 50 MS (FOR THE WORST CASE).  IF R/W/S RDY
                                ;DOES NOT SET WITHIN 50 MS, THEN IT IS ASSUMED THAT A 'SIN'
                                ;IS POSSIBLE AND THE PROGRAM WAITS FOR 1450 MS MORE, SO THAT
                                ;THE 'SIN' CAN SET.  IF 'SIN' DOES NOT SET WITHIN THIS
                                ;TIME AN ERROR IS REPORTED:
                                ;       SIN DIDN'T OCCUR
                                ;IF R/W/S RDY SETS WITHIN 50 MS THE PROGRAM PROCEEDS TO
                                ;CHECK THE NEXT DRIVE.

017652  012705  164220          WATSK:  MOV     #-6000.,R5      ;SET COUNT TO WAIT FOR
                                                                ;50 MS
```

```
3187  017656  032777  000100  161500  1$:     BIT     #RWS,@RKDS      ;R/W/S. RDY SET?
3188  017664  001042                          BNE     3$              ;YES
3189  017666  005205                          INC     R5              ;WAIT
3190  017670  001372                          BNE     1$
3191                                                                  ;50 MS OVER, R/W/S RDY
3192                                                                  ;DIDN'T SET. WAIT FOR
3193                                                                  ;SIN TO SET.
3194  017672  005004                          CLR     R4
3195  017674  012705  177777                  MOV     #177777,R5      ;SET UP COUNT
3196  017700  032777  001000  161456  2$:     BIT     #SIN,@RKDS      ;SIN SET?
3197  017706  001045                          BNE     SINST           ;YES
3198  017710  005305                          DEC     R5              ;WAIT
3199  017712  001372                          BNE     2$
3200  017714  005704                          TST     R4
3201  017716  001002                          BNE     4$
3202  017720  005204                          INC     R4
3203  017722  000766                          BR      2$
3204                                                                  ;1500 MS ELAPSED, BUT SIN
3205                                                                  ;DIDN'T SET. ERROR!
3206  017724                          4$:
3207  017724  104401  017732                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3208  017730  000415                          BR      64$             ;;GET OVER THE ASCIZ
3209                                  ;;65$:  .ASCIZ  <15><12>/SIN DIDN'T OCCUR, DRVE/
3210  017764                          64$:
3211  017764  010146                          MOV     R1,-(SP)                ;TYPE DRIVE #
3212  017766  104402                          TYPOC
3213  017770  000501                          BR      DN1DRV
3214  017772  032710  000002          3$:     BIT     #BIT1,(R0)      ;DID R/W/S RDY SET LAST TIME?
3215  017776  001476                          BEQ     DN1DRV          ;YES
3216  020000  042710  000002                  BIC     #BIT1,(R0)      ;CLR FLAG INDICATING THAT SEEK IS OK
3217  020004  104401  020272                  TYPE    ,EM1            ;REPORT THAT SEEK IS OK
3218  020010  010146                          MOV     R1,-(SP)
3219  020012  104402                          TYPOC
3220  020014  104401  020423                  TYPE    ,EM9
3221  020020  000465                          BR      DN1DRV
3222
3223                                  ;IF SIN SET, DO DRIVE RESET AND CLEAR IT
3224
3225  020022  032710  000002          SINST:  BIT     #BIT1,(R0)      ;DID 'SIN' SET LAST TIME ALSO?
3226  020026  001010                          BNE     4$              ;YES, NOTHING TO REPORT
3227  020030  052710  000002                  BIS     #BIT1,(R0)      ;SET FLAG INDICATING THAT
3228  020034  104401  020272                  TYPE    ,EM1            ;'SIN' SET, AND REPORT THE CHANGE
3229  020040  010146                          MOV     R1,-(SP)
3230  020042  104402                          TYPOC
3231  020044  104401  020415                  TYPE    ,EM8            ;TYPE 'SIN'
3232
3233  020050  017705  161322          4$:     MOV     @RKDA,R5        ;SAVE RKDA
3234  020054  012777  000001  161306          MOV     #1,@RKCS        ;DO CONTROL RESET
3235  020062  105777  161302          1$:     TSTB    @RKCS           ;WAIT FOR CONTROL RDY
3236  020066  100375                          BPL     1$
3237  020070  010577  161302                  MOV     R5,@RKDA
3238  020074  012777  000015  161266          MOV     #15,@RKCS       ;DO DRIVE RESET. RKDA
3239                                                                  ;ALREADY HAS THE DRVE #
3240  020102  105777  161262          2$:     TSTB    @RKCS           ;WAIT FOR CNTRL RDY
3241  020106  100375                          BPL     2$
3242  020110  005005                          CLR     R5
```

```
3243  020112  032777  000100  161244  3$:     BIT     #RWS,@RKDS      ;R/W/S SET?
3244  020120  001025                          BNE     DN1DRV          ;YES
3245  020122  005205                          INC     R5
3246  020124  001372                          BNE     3$              ;WAIT FOR R/W/S RDY
3247                                                                  ;REPORT ERROR. R/W/S RDY CLR
3248  020126  104401  020134                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3249  020132  000411                          BR      64$             ;;GET OVER THE ASCIZ
3250                                ;;65$:    .ASCIZ  <15><12>/RWS RDY NOT SET/
3251  020156                      64$:
3252
3253  020156  000406                          BR      DN1DRV
3254
3255                                ;IF DPL SET CLEAR THE ERROR BY DOING CONTROL RESET.
3256
3257  020160  012777  000001  161202  CLRDPL: MOV     #1,@RKCS        ;CONTROL RESET
3258  020166  105777  161176          1$:     TSTB    @RKCS           ;WAIT FOR CNTRL RDY
3259  020172  100375                          BPL     1$
3260                                ;AT THIS STAGE THE DRIVE (# IN RKDA) HAS BEEN CHECKED
3261                                ;FOR DRY, WPS, DPL, & SIN. THE POINTERS ARE INCREMENTED
3262                                ;AND THE SAME CHECKS WILL BE DONE ON THE NEXT
3263                                ;DRIVE & THEN THE NEXT ONE & SO ON. NOTE THAT
3264                                ;THIS SUB-PROGRAM KEEPS ON CYCLING THROUGH
3265                                ;ALL THE DRIVES. AT THE TIME OF ENTRY (HERE)
3266                                ;R0 POINTS TO THE FLAG FOR THE DRIVE THAT WAS
3267                                ;JUST CHECKED. BEFORE GOING ON TO THE NEXT
3268                                ;DRIVE THE HEADS ARE BROUGHT BACK TO CYLINDER
3269                                ;0 (FOR THE NEXT CYCLE).
3270
3271  020174  011077  161176          DN1DRV: MOV     (R0),@RKDA      ;GET DRIVE #
3272  020200  105777  161160                  TSTB    @RKDS           ;DRIVE PRESENT?
3273  020204  100017                          BPL     3$              ;NO
3274  020206  042777  017777  161162          BIC     #17777,@RKDA    ;CYL ADRES = 0
3275  020214  012777  000011  161146          MOV     #11,@RKCS       ;GO, SEEK
3276
3277  020222  105777  161142          1$:     TSTB    @RKCS           ;WAIT FOR CNTRL RDY
3278  020226  100375                          BPL     1$
3279
3280  020230  004737  005554                  JSR     PC,SMTME
3281  020234  032777  000100  161122  4$:     BIT     #RWS,@RKDS
3282  020242  001774                          BEQ     4$
3283
3284  020244  005720                  3$:     TST     (R0)+           ;INCREMENT POINTERS TO
3285  020246  005201                          INC     R1              ;NEXT DRIVE
3286  020250  020127  000010                  CMP     R1,#10          ;ALL DONE, THIS CYCLE?
3287  020254  001402                          BEQ     2$              ;YES
3288  020256  000137  017252                  JMP     BEGCT1          ;GO DO NEXT DRIVE
3289  020262  000137  017244          2$:     JMP     BEGCT           ;RESTART THE CYCLE OVER
3290                                                                  ;AGAIN
3291
3292
3293
3294  020266  000000                  SHFCNT: .WORD   0
3295  020270  000000                  DRVCNT: .WORD   0
3296                                ;MESSAGES
3297  020272  005015  051104  053111  EM1:    .ASCIZ  <15><12>/DRIVE /
3298  020300  020105     000
```

```
3299  020303     040  047440  020116  EM2:    .ASCIZ  /  ON LINE/
3300  020310  044514  042516     000
3301  020315     040  047440  043106  EM3:    .ASCIZ  /  OFF LINE/
3302  020322  046040  047111  000105
3303  020330  020040  051127  020124  EM4:    .ASCIZ  /  WRT PROT/
3304  020336  051120  052117     000
3305  020343     040  053440  052122  EM5:    .ASCIZ  /  WRT ENABLED/
3306  020350  042440  040516  046102
3307  020356  042105     000
3308  020361     040  042040  044522  EM6:    .ASCIZ  /  DRIVE POWER LO/
3309  020366  042526  050040  053517
3310  020374  051105  046040  000117
3311  020402  020040  047520  042527  EM7:    .ASCIZ  /  POWER UP/
3312  020410  020122  050125     000
3313  020415     040  051440  047111  EM8:    .ASCIZ  /  SIN/
3314  020422     000
3315  020423     040  051440  042505  EM9:    .ASCIZ  /  SEEK OK/
3316  020430  020113  045517     000
3317
3318                                   ;DRIVE FLAGS FOR CONTROL PANEL TEST #2
3319                                   ;BITS 15,14,13 GIVE THE DRIVE NO (EX: 0,1,2,---)
3320                                   ;BIT 7 IS SET WHEN 'DRY' BIT IS SET FOR THE DRIVE (ON LINE)
3321                                   ;BIT 7 IS CLEAR WHEN 'DRY' IS CLEAR (WHEN DRIVE IS IN LOAD/OFF LINE,
3322                                   ;DRIVE POWER IS CUT OFF)
3323                                   ;BIT 6 IS SET IF A DRIVE IS FOUND TO BE PRESENT (DRY) AT
3324                                   ;THE BEGINING. UNLIKE BIT 7 THIS BIT DOES NOT GET SET OR
3325                                   ;CLEARED AS THE DRIVE CONDITIONS CHANGE. IT JUST INDICATES
3326                                   ;THAT THE DRIVE IS AVAILABLE FOR CHECKING.
3327                                   ;BIT 0 IS SET IF 'DPL' BIT GETS SET, IE: DRIVE POWER OFF
3328                                   ;BIT 0 IS CLEARED WHEN DRIVE POWER IS ON.
3329                                   ;BIT 1 IS SET WHE, SEEK INCOMPLETE 'SIN' OCCURS.
3330                                   ;BIT 1 IS CLEAR WHEN SEEK IS OK.
3331                                   ;BIT 2 IS SET WHEN WRT PROT IS SET FROM CONSOLE.
3332                                   ;BIT 2 IS CLEARED WHEN DRIVE IS WRITE ENABLED.
3333
3334          020436                         .EVEN
3335  020436  000000                 DRIV0:  .WORD   0               ;DRIVE FLAGS
3336  020440  000000                 DRIV1:  .WORD   0
3337  020442  000000                 DRIV2:  .WORD   0
3338  020444  000000                 DRIV3:  .WORD   0
3339  020446  000000                 DRIV4:  .WORD   0
3340  020450  000000                 DRIV5:  .WORD   0
3341  020452  000000                 DRIV6:  .WORD   0
3342  020454  000000                 DRIV7:  .WORD   0
3343
3344
```

```
3345                                            .SBTTL  HEAD ALIGNMENT ROUTINE
3346
3347                                    ;HEAD ALIGNMENT ROUTINE
3348                                    ;THIS MAINTAINANCE ROUTINE IS HELPFUL IN HEAD ALIGNMENT. UPON ENTRY
3349                                    ;THE QUESTION - DRIVE? - IS ASKED , THE USER SHOULD REPLY WITH THE
3350                                    ;DRIVE NUMBER THAT IS TO BE ALIGNED. IF THE DRIVE IS AN RK-05F
3351                                    ;THE LETTER 'F' IS ADDED AS A SUFFIX. FOR SELECTING SURFACE 0
3352                                    ;PUT SW0=0, FOR SELECTING SURFACE 1 PUT SW0=1. SET SW1 =1 TO SELECT
3353                                    ;CYLINDER 64.  SET SW1=0 TO SELECT CYLINDER 105.
3354                                    ;IF THE DRIVE IS AN RK-05F, CYLINDER 64 BECOMES CYLINDER 130
3355                                    ;OF THE EVEN DRIVE, AND CYLINDER 105 BECOMES CYLINDER 5 OF THE ODD DRIVE
3356                                    ;THE HEADS ARE PLACED ON THE SELECTED CYLINDER AND DATA IS READ
3357                                    ;CONTINUOSLY FROM THE CYLINDER (SECTOR 0)
3358                                    ;  THE UPPER OR LOWER HEAD AND CYLINDER CAN BE SELECTED
3359                                    ;DYNAMICALLY, IE. THE PROGRAM DOES NOT HAVE TO BE STOPPED TO SELECT THE
3360                                    ;UPPER OR LOWER HEAD OR CYLINDER.
3361                                    ;IN ORDER TO SELECT ANOTHER DRIVE, PUT ANY SWITCH FROM SW2 TO SW15 UP AND
3362                                    ;THE PROGRAM WILL AGAIN ASK THE QUESTION (DRIVE?).
3363
3364  020456  000240                SECT.5: NOP
3365  020460  104401  020466                TYPE    ,65$            ;;TYPE ASCIZ STRING
3366  020464  000426                        BR      64$             ;;GET OVER THE ASCIZ
3367                                ;;65$:  .ASCIZ  <15><12>/SET SW0=0 FOR SURFACE 0, SW0=1 FOR SUR 1./
3368  020542                        64$:
3369  020542  104401  020550                TYPE    ,67$            ;;TYPE ASCIZ STRING
3370  020546  000432                        BR      66$             ;;GET OVER THE ASCIZ
3371                                ;;67$:  .ASCIZ  <15><12>/SET SW1=0 FOR CYLINDER 105, SW1=1 FOR CYLINDER 64/
3372  020634                        66$:
3373  020634  104401  020642                TYPE    ,69$            ;;TYPE ASCIZ STRING
3374  020640  000427                        BR      68$             ;;GET OVER THE ASCIZ
3375                                ;;69$:  .ASCIZ  <15><12>/PUT ANY SW FROM 2-15 HI TO SELECT NEW DRVE/
3376  020720                        68$:
3377  020720  104401  021336        HDALGN: TYPE    ,EM10                   ;ASK FOR DRIVE #
3378  020724  005037  021332                CLR     FFLAG                   ;FLAG FOR RK-05F
3379  020730  104406                        RDLIN                           ;GET OPR INPUT
3380  020732  012601                        MOV     (SP)+,R1                ;ADDR OF COMMAND STRING
3381  020734  112100                        MOVB    (R1)+,R0                ;FIRST CHAR
3382  020736  162700  000060                SUB     #60,R0                  ;0 TO 7
3383  020742  002766                        BLT     HDALGN                  ;TOO SMALL
3384  020744  022700  000067                CMP     #67,R0                  ;MUST BE 7 OR LESS
3385  020750  002763                        BLT     HDALGN                  ;TOO BIG
3386  020752  000241                        CLC
3387  020754  006000                        ROR     R0
3388  020756  006000                        ROR     R0
3389  020760  006000                        ROR     R0
3390  020762  006000                        ROR     R0
3391  020764  010037  020436                MOV     R0,DRIV0                ;ADDRESS OF DRIVE
3392  020770  112100                        MOVB    (R1)+,R0                ;NEXT INPUT CHAR
3393  020772  001412                        BEQ     5$                      ;ALL DONE IF C.R.
3394  020774  020027  000106                CMP     R0,#'F                  ;IS IT F?
3395  021000  001347                        BNE     HDALGN                  ;NO, SO ERROR
3396  021002  105711                        TSTB    (R1)                    ;NEXT CHAR MUST BE C.R.
3397  021004  001345                        BNE     HDALGN                  ;ELSE, ERROR
3398  021006  042737  020000  020436        BIC     #BIT13,DRIV0            ;USE EVEN DRIVE IF RK-05F
3399  021014  005237  021332                INC     FFLAG                   ;SHOW F TYPE DRIVE
3400  021020  013700  020436        5$:     MOV     DRIV0,R0                ;DRIVE ADDR TO R0
```

```
3401  021024  017737  160110  021334          MOV     @SWR,SWTCH              ;HOLD SWITCHES
3402  021032  042737  177775  021334          BIC     #177775,SWTCH           ;WANT SW1 ONLY
3403  021040  001005                          BNE     7$                      ;SW1 SET, SO LOW CYLINDER
3404  021042  005737  021332                  TST     FFLAG                   ;F DRIVE?
3405  021046  001402                          BEQ     7$                      ;NO
3406  021050  052700  020000                  BIS     #BIT13,R0               ;ODD DRIVE IF HIGH TRACK OF F
3407  021054  010077  160316          7$:     MOV     R0,@RKDA                ;ADDRESS DRIVE
3408  021060  012777  000017  160302          MOV     #17,@RKCS               ;WRITE PROTECT
3409  021066  105777  160276          8$:     TSTB    @RKCS
3410  021072  100375                          BPL     8$                      ;WAIT FOR DRIVE READY
3411  021074  012777  000001  160266          MOV     #1,@RKCS                ;RESET CONTROLLER
3412  021102  105777  160262          9$:     TSTB    @RKCS
3413  021106  100375                          BPL     9$                      ;WAIT FOR READY
3414  021110  005737  021332                  TST     FFLAG                   ;F DRIVE?
3415  021114  001410                          BEQ     13$                     ;NO
3416  021116  012701  000240                  MOV     #5.*40,R1               ;TRACK 5 OF HIGH
3417  021122  005737  021334                  TST     SWTCH                   ;SW1 SET?
3418  021126  001412                          BEQ     10$                     ;YES SO TEST TRACK 8 OF DRIVE HIGH
3419  021130  062701  010100                  ADD     #130.*40,R1             ;TRACK 130. IF SW1 SET
3420  021134  000407                          BR      10$
3421  021136  012701  004000          13$:    MOV     #64.*40,R1              ;CYLINDER 64 IF NOT F
3422  021142  005737  021334                  TST     SWTCH                   ;SW1 SET?
3423  021146  001002                          BNE     10$                     ;YES, SO CYLINDER 64
3424  021150  062701  002440                  ADD     #41.*40,R1              ;CYLINDER 105
3425  021154  005777  160210          10$:    TST     @RKCS                   ;ANY ERROR?
3426  021160  100006                          BPL     11$                     ;NO, CONTINUE
3427  021162  012777  000001  160200          MOV     #1,@RKCS                ;RESET
3428  021170  105777  160174          12$:    TSTB    @RKCS
3429  021174  100375                          BPL     12$                     ;WAIT FOR READY
3430  021176  017702  157736          11$:    MOV     @SWR,R2                 ;SWITCH REG TO R2
3431  021202  042702  177775                  BIC     #177775,R2              ;SW1 ONLY
3432  021206  020237  021334                  CMP     R2,SWTCH                ;ANY CHANGE SINCE LAST?
3433  021212  001302                          BNE     5$                      ;YES, GO SET-UP ADDR AGAIN
3434  021214  010077  160156          6$:     MOV     R0,@RKDA                ;ADDRESS THE DRIVE
3435  021220  012777  000017  160142          MOV     #17,@RKCS               ;WRITE  PROTECT THE DRIVE
3436  021226  105777  160136                  TSTB    @RKCS                   ;WAIT FOR CONTROL RDY
3437  021232  100375                          BPL     .-4
3438  021234  042700  000020          4$:     BIC     #20,R0                  ;CLEAR TRACK ADDR
3439  021240  032777  000001  157672          BIT     #1,@SWR                 ;SW0 SET?
3440  021246  001402                          BEQ     2$                      ;NO TEST TRACK 0
3441  021250  052700  000020                  BIS     #20,R0                  ;TEST TRACK 1
3442  021254  042700  017740          2$:     BIC     #17740,R0               ;CLEAR CYLINDER ADDR
3443  021260  050100                          BIS     R1,R0                   ;PUT CYLINDER ADDR IN ADDR
3444  021262  010077  160110                  MOV     R0,@RKDA                ;ADRES THE DRIVE
3445  021266  012777  177400  160076          MOV     #-400,@RKWC             ;READ 1 SECTOR
3446  021274  012777  007316  160072          MOV     #RDBUFF,@RKBA           ;INTO THIS BUFFER
3447  021302  012777  000005  160060          MOV     #5,@RKCS                ;READ, GO
3448
3449  021310  105777  160054          3$:     TSTB    @RKCS                   ;DONE?
3450  021314  100375                          BPL     3$                      ;NO
3451
3452  021316  032777  177774  157614          BIT     #177774,@SWR            ;EXIT OUT?
3453  021324  001713                          BEQ     10$                     ;NO, CONTINUE ON THIS DRIVE
3454  021326  000137  020720                  JMP     HDALGN                  ;YES, GET NEW DRIVE
3455
3456  021332  000000          FFLAG:  0
```

```
                                                SWTCH:  0
                                EM10:   .ASCIZ  15\<12> DRIVE?/
                                        .000


                                        .SBTTL  DISK POWER FAILURE TEST

                                ;(DISK)POWER FAILURE (DURING DISK WRITE) TEST
                                ;THIS TEST CHECKS THAT THE INFORMATION WRITTEN ON THE DISK IS
                                ;NOT DESTROYED WHEN THE DISK SENSES A LOSS OF POWER WHILE DOING A WRITE
                                ;AND RETRACTS THE HEADS. UPON ENTRY THE PROGRAM FINDS OUT THE
                                ;FIRST AVAILABLE DRIVE INDICATES IT (DRIVE #X) AND PROCEEDS TO TEST.
                                ;CYLINDERS 0 TO 15 ARE WRITTEN WITH UNIQUE PATTERNS, THEN THE HEADS
                                ;ARE POSITIONED ON CYLINDER 10 (DECIMAL) AND A MESSAGE (DROP
                                ;POWER) IS GIVEN.  AFTER RECEIVING THIS MESSAGE THE USER SHOULD
                                ;DROP POWER FROM THE DRIVE.  ON SENSING A LOSS OF POWER, THE
                                ;PROGRAM ASKS THE USER TO PUT BACK THE POWER.  THE ERRORS (DPL)
                                ;ARE CLEARED AND A WRITE-CHECK IS PERFORMED TO CHECK IF THE
                                ;UNIQUE PATTERNS ON THE DISK (CYLINDERS 0-9 AND 11-15) ARE STILL
                                ;THERE.  IF NOT A WRITE CHECK ERROR IS REPORTED.
                        021350          .EVEN

021350  005000          SECT.6: CLR     R0              ;INITIALIZE DRIVE #
021352  005037  020436          CLR     DRIVO
021356  010077  160014  8$:     MOV     R0,@RKDA        ;IS IT PRESENT?
021362  105777  157776          TSTB    @RKDS           ;IF NOT SKIP
021366  100436                  BMI     9$
021370  005237  020436  10$:    INC     DRIVO
021374  062700  020000          ADD     #20000,R0
021400  001366                  BNE     8$
021402  000762                  BR      SECT.6
021404  104401  020272  9$:     TYPE    ,EM1
021410  013746  020436          MOV     DRIVO,-(SP)     ;GET DRIVE #
021414  104402                  TYPOC
021416  012777  000001  157744  MOV     #1,@RKCS        ;CONTROL RESET
021424  105777  157740          TSTB    @RKCS
021430  100375                  BPL     .-4
021432  005005                  CLR     R5              ;INITIALIZE PATTERN TO BE WRITTEN
                                                        ;0 ON CYL 0, 1 ON CYL 1, ETC
021434  013701  001376          MOV     RKDA,R1
021440  013702  001372          MOV     RkWC,R2
021444  013703  001374          MOV     RkBA,R3
021450  013704  001370          MOV     RKCS,R4
021454  010011                  MOV     R0,@R1
021456  010537  022006  1$:     MOV     R5,BUFR         ;FILL THE PATTER IN DATA BUFFER.
021462  012713  022006          MOV     #BUFR,@R3       ;BUS ADRES
021466  012712  164000          MOV     #-14000,@R2     ;WRITE 1 CYL (256X12X2 WORDS)
021472  012714  004003          MOV     #4003,@R4       ;WRITE, GO, IBA SET
021476  105714                  TSTB    @R4             ;WAIT FOR CONTROL READY
021500  100376                  BPL     .-2
                                                        ;DONE
021502  005205                  INC     R5              ;WRITTEN ALL 15 CYLINDERS?
021504  020527  000020          CMP     R5,#20          ;IF NOT, GO BAK
021510  001362                  BNE     1$
```

```
 3513  021512  010005                          MOV     R0,R5           ;DRIVE #
 3514  021514  052705  000500                  BIS     #500,R5         ;CYL 10
 3515  021520  012737  000012  022006          MOV     #12,BUFR        ;PATERN TO BE WRITTEN
 3516  021526  010511                          MOV     R5,@R1          ;ADRES THE DISK
 3517  021530  012712  164000                  MOV     #-14000,@R2     ;WORD COUNT= 1 CYLINDER
 3518  021534  012713  022006                  MOV     #BUFR,@R3       ;BUS ADRES
 3519  021540  012714  004003                  MOV     #4003,@R4       ;WRITE, GO, IBA
 3520  021544  032777  000100  157612  2$:     BIT     #RWS,@RKDS      ;WAIT FOR THE HEADS TO SETTLE
 3521  021552  001774                          BEQ     2$              ;ON CYL 10
 3522  021554  104401  021562                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 3523  021560  000406                          BR      64$             ;;GET OVER THE ASCIZ
 3524                                  ;;65$:  .ASCIZ  /DROP POWER/
 3525  021576                          64$:
 3526
 3527  021576  000407                          BR      5$
 3528  021600  010511                  3$:     MOV     R5,@R1          ;ADRES THE DISK
 3529  021602  012712  164000                  MOV     #-14000,@R2     ;WORD COUNT= 1 CYLINDER
 3530  021606  012713  022006                  MOV     #BUFR,@R3       ;BUS ADRES
 3531  021612  012714  004003                  MOV     #4003,@R4       ;WRITE, GO, IBA
 3532  021616  105714                  5$:     TSTB    @R4             ;WAIT FOR CONTROL READY
 3533  021620  100376                          BPL     .-2
 3534  021622  005714                          TST     @R4             ;WAIT FOR DRIVE POWER TO GO DOWN,
 3535                                                                  ;OTHERWISE, KEEP ON WRITING ON CYL 10
 3536  021624  100365                          BPL     3$
 3537
 3538                                                                  ;IF DRIVE POWER LOSS WAS SENSED,
 3539                                                                  ;ASK TO PUT POWER ON.
 3540  021626  104401  021634                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 3541  021632  000406                          BR      66$             ;;GET OVER THE ASCIZ
 3542                                  ;;67$:  .ASCIZ  <15><12>/POWER ON/
 3543  021650                          66$:
 3544  021650  105777  157510                  TSTB    @RKDS           ;WAIT FOR DRIVE READY
 3545  021654  100375                          BPL     .-4
 3546  021656  012714  000001                  MOV     #1,@R4          ;CONTROL RESET, CLEAR ERROR
 3547  021662  105714                          TSTB    @R4
 3548  021664  100376                          BPL     .-2
 3549  021666  010077  157504                  MOV     R0,@RKDA
 3550  021672  005005                          CLR     R5              ;INITIALIZE PATTERN
 3551  021674  010537  022006          6$:     MOV     R5,BUFR
 3552  021700  012713  022006                  MOV     #BUFR,@R3
 3553  021704  012712  164000                  MOV     #-14000,@R2
 3554  021710  012714  004007                  MOV     #4007,@R4       ;WRITE CHECK, GO, IBA
 3555  021714  105714                          TSTB    @R4
 3556  021716  100376                          BPL     .-2
 3557
 3558  021720  005714                          TST     @R4             ;ANY ERROR?
 3559  021722  100023                          BPL     7$              ;NO
 3560  021724  104401  021732                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 3561  021730  000416                          BR      68$             ;;GET OVER THE ASCIZ
 3562                                  ;;69$:  .ASCIZ  <15><12>/ERROR, ON PWR-UP,  RKDA=/
 3563  021766                          68$:
 3564  021766  011146                          MOV     @R1,-(SP)
 3565  021770  104402                          TYPOC
 3566  021772  005205                  7$:     INC     R5
 3567  021774  020527  000020                  CMP     R5,#20
 3568  022000  001335                          BNE     6$
```

```
3569  022002  000137  021370                  JMP     10$
3570
3571  022006  000000                  BUFR:   .WORD   0
3572
3573
3574                                  .SBTTL  TYPE ROUTINE
3575
3576                                  ;;*****************************************************************
3577                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TEPMINATE WITH A 0 BYTE.
3578                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3579                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3580                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTEPS REQUIRED.
3581                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3582                                  ;*
3583                                  ;*CALL:
3584                                  ;*1) USING A TRAP INSTRUCTION
3585                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3586                                  ;*OR
3587                                  ;*      TYPE
3588                                  ;*      MESADR
3589                                  ;*
3590
3591  022010  105737  001157  $TYPE:   TSTB    $TPFLG          ;;IS THERE A TERMINAL?
3592  022014  100002                  BPL     1$              ;;BR IF YES
3593  022016  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
3594  022020  000407                  BR      3$              ;;LEAVE
3595  022022  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
3596  022024  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
3597  022030  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
3598  022032  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
3599  022034  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
3600  022036  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
3601  022040  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
3602  022044  000002                  RTI                     ;;RETURN
3603  022046  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
3604  022052  001430                  BEQ     8$
3605  022054  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
3606  022060  001006                  BNE     5$
3607  022062  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
3608  022064  104401                  TYPE                    ;;TYPE A CR AND LF
3609  022066  001161                  $CRLF
3610  022070  105037  022224          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
3611  022074  000755                  BR      2$              ;;GET NEXT CHARACTER
3612  022076  004737  022160  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
3613  022102  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
3614  022106  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
3615  022110  013746  001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
3616                                                          ;;AND THE NULL CHAR.
3617  022114  105366  000001  7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
3618  022120  002770                  BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
3619  022122  004737  022160          JSR     PC,$TYPEC       ;;GO TYPE A NULL
3620  022126  105337  022224          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
3621  022132  000770                  BR      7$              ;;LOOP
3622
3623                                  ;HORIZ  TAL TAB PROCESSOR
3624
```

```
3625  022134  112716  000040          9$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3626  022140  004737  022160          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
3627  022144  132737  000007  022224          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3628  022152  001372                          BNE     9$              ;;TAB STOP
3629  022154  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
3630  022156  000724                          BR      2$              ;;GET NEXT CHARACTER
3631  022160  105777  156764        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
3632  022164  100375                          BPL     $TYPEC
3633  022166  116677  000002  156756        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3634  022174  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3635  022202  001003                          BNE     1$              ;;BRANCH IF NO
3636  022204  105037  022224                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3637  022210  000406                          BR      $TYPEX          ;;EXIT
3638  022212  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3639  022220  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
3640  022222  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
3641  022224  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
3642  022226  000207          $TYPEX: RTS     PC
3643
3644                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3645
3646                                  ;;**************************************************************
3647                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3648                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3649                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3650                                  ;*CALL:
3651                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3652                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
3653                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3654                                  ;*      .BYTE   M               ;;M=1 OR 0
3655                                  ;*                                      ;;1=TYPE LEADING ZEROS
3656                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
3657                                  ;*
3658                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3659                                  ;*$TYPOS OR $TYPOC
3660                                  ;*CALL:
3661                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3662                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
3663                                  ;*
3664                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3665                                  ;*CALL:
3666                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3667                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
3668
3669  022230  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
3670  022234  116637  000001  022453          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
3671  022242  112637  022455                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3672  022246  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3673  022252  000406                          BR      $TYPON
3674  022254  112737  000001  022453  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3675  022262  112737  000006  022455          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3676  022270  112737  000005  022452  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3677  022276  010346                          MOV     R3,-(SP)        ;;SAVE R3
3678  022300  010446                          MOV     R4,-(SP)        ;;SAVE R4
3679  022302  010546                          MOV     R5,-(SP)        ;;SAVE R5
3680  022304  113704  022455                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
```

```
3681  022310  005404                          NEG     R4
3682  022312  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3683  022316  110437  022454                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3684  022322  113704  022453                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3685  022326  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3686  022332  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
3687  022334  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3688  022336  000404                          BR      3$              ;;GO DO MSB
3689  022340  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
3690  022342  006105                          ROL     R5
3691  022344  006105                          ROL     R5
3692  022346  010503                          MOV     R5,R3
3693  022350  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3694  022352  105337  022454                  DECB    $OMODE          ;;TYPE THIS DIGIT?
3695  022356  100016                          BPL     7$              ;;BR IF NO
3696  022360  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
3697  022364  001002                          BNE     4$              ;;TEST FOR 0
3698  022366  005704                          TST     R4              ;;SUPPRESS THIS 0?
3699  022370  001403                          BEQ     5$              ;;BR IF YES
3700  022372  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3701  022374  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3702  022400  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3703  022404  110337  022450                  MOVB    R3,8$           ;;SAVE FOR TYPING
3704  022410  104401  022450                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
3705  022414  105337  022452          7$:     DECB    $OCNT           ;;COUNT BY 1
3706  022420  003347                          BGT     2$              ;;BR IF MORE TO DO
3707  022422  002402                          BLT     6$              ;;BR IF DONE
3708  022424  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3709  022426  000744                          BR      2$              ;;GO DO THE LAST DIGIT
3710  022430  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
3711  022432  012604                          MOV     (SP)+,R4        ;;RESTORE R4
3712  022434  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3713  022436  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3714  022444  012616                          MOV     (SP)+,(SP)
3715  022446  000002                          RTI                     ;;RETURN
3716  022450     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3717  022451     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3718  022452     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3719  022453     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3720  022454  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3721                                  .SBTTL  TTY INPUT ROUTINE
3722
3723                                  ;;**************************************************************
3724                                  .ENABL  LSB
3725
3726                                  .DSABL  LSB
3727
3728
3729                                  ;;**************************************************************
3730                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3731                                  ;*CALL:
3732                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
3733                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
3734                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
3735                                  ;
3736
```

```
3737  022456  011646                      $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
3738  022460  016666  000004  000002              MOV     4(SP),2(SP)     ;;SAVE THE PS
3739  022466  105777  156452              1$:     TSTB    @$TKS           ;;WAIT FOR
3740  022472  100375                              BPL     1$              ;;A CHARACTER
3741  022474  117766  156446  000004              MOVB    @$TKB,4(SP)     ;;READ THE TTY
3742  022502  042766  177600  000004              BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
3743  022510  026627  000004  000023              CMP     4(SP),#23       ;;IS IT A CONTROL-S?
3744  022516  001013                              BNE     3$              ;;BRANCH IF NO
3745  022520  105777  156420              2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
3746  022524  100375                              BPL     2$              ;;LOOP UNTIL ITS THERE
3747  022526  117746  156414                      MOVB    @$TKB,-(SP)     ;;GET CHARACTER
3748  022532  042716  177600                      BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
3749  022536  022627  000021                      CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
3750  022542  001366                              BNE     2$              ;;IF NOT DISCARD IT
3751  022544  000750                              BR      1$              ;;YES, RESUME
3752  022546  026627  000004  000140  3$:         CMP     4(SP),#140      ;;IS IT UPPER CASE?
3753  022554  002407                              BLT     4$              ;;BRANCH IF YES
3754  022556  026627  000004  000175              CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
3755  022564  003003                              BGT     4$              ;;BRANCH IF YES
3756  022566  042766  000040  000004              BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
3757  022574  000002                      4$:     RTI                     ;;GO BACK TO USER
3758                                      ;;**************************************************************
3759                                      ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
3760                                      ;*CALL:
3761                                      ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
3762                                      ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3763                                      ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
3764
3765  022576  010346                      $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
3766  022600  005046                              CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
3767  022602  012703  023032              1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
3768  022606  022703  023062              2$:     CMP     #$TTYIN+30,R3   ;;BUFFER FULL?
3769  022612  101456                              BLOS    4$              ;;BR IF YES
3770  022614  104405                              RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
3771  022616  112613                              MOVB    (SP)+,(R3)      ;;GET CHARACTER
3772  022620  122713  000177              10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
3773  022624  001022                              BNE     5$              ;;BR IF NO
3774  022626  005716                              TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
3775  022630  001007                              BNE     6$              ;;BR IF NO
3776  022632  112737  000134  023030              MOVB    #'\,9$          ;;TYPE A BACK SLASH
3777  022640  104401  023030                      TYPE    ,9$
3778  022644  012716  177777                      MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
3779  022650  005303                      6$:     DEC     R3              ;;BACKUP BY ONE
3780  022652  020327  023032                      CMP     R3,#$TTYIN      ;;STACK EMPTY?
3781  022656  103434                              BLO     4$              ;;BR IF YES
3782  022660  111337  023030                      MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
3783  022664  104401  023030                      TYPE    ,9$             ;;GO TYPE
3784  022670  000746                              BR      2$              ;;GO READ ANOTHER CHAR.
3785  022672  005716                      5$:     TST     (SP)            ;;RUBOUT KEY SET?
3786  022674  001406                              BEQ     7$              ;;BR IF NO
3787  022676  112737  000134  023030              MOVB    #'\,9$          ;;TYPE A BACK SLASH
3788  022704  104401  023030                      TYPE    ,9$
3789  022710  005016                              CLR     (SP)            ;;CLEAR THE RUBOUT KEY
3790  022712  122713  000025              7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
3791  022716  001003                              BNE     8$              ;;BR IF NO
3792  022720  104401  023062                      TYPE    .$CNTLU         ;;TYPE A CONTROL "U"
```

```
 3793  022724  000726                          BR      1$              ;;GO START OVER
 3794  022726  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
 3795  022732  001011                          BNE     3$              ;;BRANCH IF NO
 3796  022734  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
 3797  022736  104401  001161                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
 3798  022742  104401  023032                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
 3799  022746  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
 3800  022750  104401  001160          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 3801  022754  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 3802  022756  111337  023030          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 3803  022762  104401  023030                  TYPE    ,9$
 3804  022766  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 3805  022772  001305                          BNE     2$              ;;LOOP IF NOT RETURN
 3806  022774  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 3807  023000  104401  001162                  TYPE    ,$LF            ;;TYPE A LINE FEED
 3808  023004  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 3809  023006  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 3810  023010  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 3811  023012  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 3812  023020  012766  023032  000004          MOV     #$TTYIN,4(SP)
 3813  023026  000002                          RTI                     ;;RETURN
 3814  023030     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 3815  023031     000                          .BYTE   0               ;;TERMINATOR
 3816  023032  000030                  $TTYIN: .BLKB   30              ;;RESERVE 30 BYTES FOR TTY INPUT
 3817  023062  052536  005015     000 $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 3818  023067     136  006507  000012 $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 3819  023074  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
 3820  023102  020075     000
 3821  023105     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
 3822  023112  036440  000040
 3823                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 3824
 3825                                  ;;*****************************************************************
 3826                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 3827                                  ;*CHANGE IT TO BINARY.
 3828                                  ;*CALL:
 3829                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 3830                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 3831                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 3832
 3833  023116  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 3834  023120  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
 3835  023126  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 3836  023130  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 3837  023132  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3838  023134  104406                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
 3839  023136  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 3840  023140  005001                          CLR     R1              ;;CLEAR DATA WORD
 3841  023142  005002                          CLR     R2
 3842  023144  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 3843  023146  001412                          BEQ     3$              ;;IF ZERO GET OUT
 3844  023150  006301                          ASL     R1              ;;*2
 3845  023152  006102                          ROL     R2
 3846  023154  006301                          ASL     R1              ;;*4
 3847  023156  006102                          ROL     R2
 3848  023160  006301                          ASL     R1              ;;*8
```

```
3849  023162  006102                          ROL     R2
3850  023164  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
3851  023170  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
3852  023172  000764                          BR      2$              ;;LOOP
3853  023174  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
3854  023176  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
3855  023202  010237  023215                  MOV     R2,$HIOCT
3856  023206  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3857  023210  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
3858  023212  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3859  023214  000002                          RTI                     ;;RETURN
3860  023216  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
3861                                  .SBTTL  TRAP DECODER
3862
3863                                  ;;**************************************************************
3864                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3865                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3866                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3867                                  ;*GO TO THAT ROUTINE.
3868
3869  023220  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
3870  023222  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
3871  023226  005740                          TST     -(R0)           ;;BACKUP BY 2
3872  023230  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
3873  023232  006300                          ASL     R0              ;;POSITION FOR INDEXING
3874  023234  016000  023254                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
3875  023240  000200                          RTS     R0              ;;GO TO ROUTINE
3876
3877
3878                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3879
3880  023242  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
3881  023244  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
3882  023252  000002                          RTI                     ;;RESTORE THE PSW
3883
3884                                  .SBTTL  TRAP TABLE
3885
3886                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3887                                  ;*BY THE "TRAP" INSTRUCTION.
3888
3889                                  ;       ROUTINE
3890                                  ;       -------
3891  023254  023242                  $TRPAD: .WORD   $TRAP2
3892  023256  022010                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
3893  023260  022254                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3894  023262  022230                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3895  023264  022270                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3896
3897
3898  023266  022456                          $RDCHR  ;;CALL=RDCHR    TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
3899  023270  022576                          $RDLIN  ;;CALL=RDLIN    TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
3900  023272  023116                          $RDOCT  ;;CALL=RDOCT    TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
3901                                  .SBTTL  POWER DOWN AND UP ROUTINES
3902
3903                                  ;;**************************************************************
3904                                  ;POWER DOWN ROUTINE
```

```
 3905  023274  012737  023434  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 3906  023302  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
 3907  023310  010046                          MOV     R0,-(SP)         ;;PUSH R0 ON STACK
 3908  023312  010146                          MOV     R1,-(SP)         ;;PUSH R1 ON STACK
 3909  023314  010246                          MOV     R2,-(SP)         ;;PUSH R2 ON STACK
 3910  023316  010346                          MOV     R3,-(SP)         ;;PUSH R3 ON STACK
 3911  023320  010446                          MOV     R4,-(SP)         ;;PUSH R4 ON STACK
 3912  023322  010546                          MOV     R5,-(SP)         ;;PUSH R5 ON STACK
 3913  023324  017746  155610                  MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
 3914  023330  010637  023440                  MOV     SP,$SAVR6        ;;SAVE SP
 3915  023334  012737  023346  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 3916  023342  000000                          HALT
 3917  023344  000776                          BR      .-2              ;;HANG UP
 3918
 3919                                  ;;*****************************************************************
 3920                                  ;POWER UP ROUTINE
 3921  023346  012737  023434  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 3922  023354  013706  023440                  MOV     $SAVR6,SP        ;;GET SP
 3923  023360  005037  023440                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
 3924  023364  005237  023440          1$:     INC     $SAVR6           ;;WAIT FOR THE INC
 3925  023370  001375                          BNE     1$               ;;OF  WORD
 3926  023372  012677  155542                  MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
 3927  023376  012605                          MOV     (SP)+,R5         ;;POP STACK INTO R5
 3928  023400  012604                          MOV     (SP)+,R4         ;;POP STACK INTO R4
 3929  023402  012603                          MOV     (SP)+,R3         ;;POP STACK INTO R3
 3930  023404  012602                          MOV     (SP)+,R2         ;;POP STACK INTO R2
 3931  023406  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
 3932  023410  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
 3933  023412  012737  023274  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 3934  023420  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
 3935  023426  104401                          TYPE                     ;REPORT THE POWER FAILURE
 3936  023430  023442                  $PWRMG: .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
 3937  023432  000002                          RTI
 3938  023434  000000                  $ILLUP: HALT                     ;;THE POWER UP SEQJENCE WAS STARTED
 3939  023436  000776                          BR      .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
 3940  023440  000000                  $SAVR6: 0                        ;;PUT THE SP HERE
 3941  023442  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
 3942  023450  000122
 3943                                  .EVEN
 3944          000001                  .END
```

```
AUTSL2  002604          1614*   1831
BA      001406          1475*   2530*   2532*   2534*   2536*   2557    2591
BADINP  003234          1675    1677    1684*
BADONE  010320          2316*
BAD.IN  011702          2470*
BAD.ON  013702          2755*   2773    2775
BASE    001266          1419*   2006
BASINC  005320          2011    2023*
BEGCT   017244          3064*   3289
BEGCT1  017252          3067*   3288
BEGIN   002562          1591    1598*
BIT0  = 000001          1323*   2249    2562    2579    2596    3154    3161    3165    3172
BIT00 = 000001          1313*   1323
BIT01 = 000002          1312*   1322
BIT02 = 000004          1311*   1321
BIT03 = 000010          1310*   1320
BIT04 = 000020          1309*   1319
BIT05 = 000040          1308*   1318
BIT06 = 000100          1307*   1317
BIT07 = 000200          1306*   1316
BIT08 = 000400          1305*   1315
BIT09 = 001000          1304*   1314
BIT1  = 000002          1322*   2560    2595    3214    3216    3225    3227
BIT10 = 002000          1303*   2559    2576
BIT11 = 004000          1302*   2559    2593
BIT12 = 010000          1301*   1489
BIT13 = 020000          1300*   2957    2967    3398    3406
BIT14 = 040000          1299*   1679    1712    1810    1844    2582    2720
BIT15 = 100000          1298*   1780    2171    2173    2282    2852    2868
BIT2  = 000004          1321*   2577    2595    3111    3117    3121    3129
BIT3  = 000010          1320*
BIT4  = 000020          1319*   1971    1973    1976    2108    2110    2113
BIT5  = 000040          1318*   1491    1925    2943
BIT6  = 000100          1317*   1490    2520    2525    2636    3044    3138
BIT7  = 000200          1316*   1761    1765    2284    3044    3074    3096
BIT8  = 000400          1315*   2593
BIT9  = 001000          1314*   1492
BPTVEC= 000014          1330*
BUFF    013620          1475    2748*
BUFR    022006          3502*   3503    3515*   3518    3530    3550*   3551    3571*
BUSADR  001336          1453*   1967*   2038    2098*
CHECK1  005410          2042*   2052    2056
CHKCNT  001350          1458*   2185*   2226*
CLRDPL  020160          3155    3163    3257*
CNTSIN  001322          1442*   1876*   2147*
COM     012214          2515*   2528
COMMON  012210          2514*
COMND   001316          1438*   1959*   2075*   2167
CONTIN  011314          2399*   2465
CONTRL  001332          1451*   1969*   2040    2101*
CR    = 000015          1238*   3634    3644
CRLF  = 000200          1239*   1539    3605    3644
CYCLE   004134          1705    1843*
CYCLE2  004160          1849*
CYCL2   004150          1845    1847*
CYL     013432          2684*   2688*   2692*   2696*
```

```
CYLADR  005242          1964    2006*   2093
CYLAD2  005246          2007*   2027
CYLBAS  001315          1437*
CYLCNT  001342          1455*   1968
CYLOFF  005252          1979    2008*   2116
CYLTBL  001274          1426*   1992    2007    2025
DA      001410          1476*   2529*   2538*   2539    2555    2572    2589    2604
DDISP = 177570          1245*   1366    1513
DISPLA  001142          1366*   1513*   1521*
DISPRE  000174          1217*   1521
DN1DRV  020174          3142    3213    3215    3221    3244    3253    3271*
DO2     004210          1717    1785    1858*
DPL   = 010000          1489*   3151
DRACTV  001164          1394*   1704*
DRCNT1  001311          1433*   1626*   1680    1825*   2105
DRIVE   001314          1436*   1862*   1881    1961    2776*   2788
DRIV0   020436          2331    2346    2366    2429    2439    2446    2457    3031    3064    3335*   3391*   3398*   3400
                        3481*   3485*   3490
DRIV1   020440          3336*
DRIV2   020442          3337*
DRIV3   020444          3338*
DRIV4   020446          3339*
DRIV5   020450          3340*
DRIV6   020452          3341*
DRIV7   020454          3342*
DRVCNT  020270          2486*   2490*   2491    2494    3295*
DRV0    001206          1398*   1638    1699    1710    1778    1826    1856    1963    2080    2091
DSKADR  001334          1452*   1869*   1901    1971    1973*   1976*   2020*   2021*   2037    2096*   2097*   2104*   2108
                        2110*   2113*   2210
DSKTMP  001352          1459*   1868*   2096    2505*   2516    2556    2573    2590    2783*   2818    2847    2912    2917
                        2924    2950    2973
DSWR  = 177570          1244*   1365    1512
ECNT    001321          1441*   1875*   2134*   2149
EMTVEC= 000030          1333*
EM1     020272          2496    3040    3075    3092    3114    3125    3157    3168    3217    3228    3297*   3489
EM10    021336          3377    3458*
EM2     020303          3043    3078    3299*
EM3     020315          3095    3301*
EM4     020330          3128    3303*
EM5     020343          3118    3305*
EM6     020361          3160    3308*
EM7     020402          3171    3311*
EM8     020415          3231    3313*
EM9     020423          3220    3315*
END     013604          2737    2740*
ERRCHK  006042          2044    2126*
ERRFLG  001360          1462*   2045    2047*   2131*   2141*
ERRRF   001424          1482*   2507*   2584*   2642*   2643
ERRRFC  001426          1483*   2508*   2608*   2649*   2652
ERRVEC= 000004          1326*   1510    1511*   1522*
ERRWCH  001430          1484*   2509*   2616*   2726*   2727
ERRWCS  001432          1485*   2510*   2617*   2722*   2723
ERRWF   001422          1481*   2506*   2567*   2623*   2624
EXECUT  005342          1970    2034*   2048    2062    2102
EXIT    003326          1700    1709*
EXITA   003614          1772*   1774    1812
```

```
EXITB   003622          1775*   1808
EXITX   003370          1711    1713    1721*
EXIT2   006036          2081    2120*
EXTER2  003332          1710*   1715    1720    2306
EXTR    001420          1480*   2513*   2561    2578    2594
FAIL    013210          2658*   2724
FAILED  013172          2653    2655*
FFLAG   021332          3378*   3399*   3404    3414    3456*
FIL.DN  003632          1778*   1781
GDI     012422          2492    2547*
GNS   = ****** U        1216    1538    1542    1546    1550    1554    1558    1562    1566    1570    1574    1579    1595
                        1617    1621    1650    1662    1669    1687    1724    1734    1742    1751    1790    1834    1879
                        1887    1891    1915    1922    1929    1949    2060    2154    2160    2165    2194    2203    2209
                        2216    2222    2259    2263    2268    2276    2292    2319    2328    2357    2361    2378    2382
                        2386    2390    2409    2473    2481    2758    2765    2786    2794    2798    2802    2808    2814
                        2827    2831    2835    2840    2844    2856    2862    2872    2878    2883    2887    2891    2896
                        2900    2904    2908    2937    2947    2961    2971    2989    2994    3002    3008    3083    3209
                        3250    3367    3371    3375    3524    3541    3561    3892    3893    3894    3895    3898    3899
                        3900
GO      003252          1683    1694*
GOOD    012376          2541*
GO1     003264          1698*   1703    1708    2175
GO2     003276          1696    1701*
GO3     003304          1702    1704*
HDALGN  020720          3377*   3383    3385    3395    3397    3454
HDRFLG  001320          1440*   2183*   2190    2196*
HT    = 000011          1236*   3603    3644
IDEX    001324          1444*   1695*   1847    1873*
ILEGAL  004102          1637    1821    1831*
INITIL  004464          1894    1900*   2057
INITI2  004476          1903*   1924    1931
INIT.2  010426          2341*
INSYS2  003466          1739*   1771
IO      012426          2531    2533    2535    2537    2554*   2638    2725
IOTVEC= 000020          1331*
KYTEMP  001330          1450*   1627*   1628*   1629    1631*   1654*   1655    1656    1658    1672*   1673    1674*   1676
                        1678*   1679*   1682    1819    1822*   1823    1824*
LDFLG   004164          1850*   2303
LDSEEK  011514          2429*   2464
LF    = 000012          1237*   3638    3644
LFLF  = 105212          1474*   2660    2661    2703    2704
LOGA    001166          1396*   1625    1682*   1694    1709    1738    1850    2076    2243    2297
MANSEL  010316          2228    2310*   2940    3012
MASK    005174          1784    1849    1987*   2092
MASKER  003770          1767    1769    1803*
MODE    001312          1226*   1434*   1494*   1524
MORE    012754          2604*   2607
MOUNT   004256          1870    1873*
MSKTBL  001256          1410*   1848    2239
NEXT    012312          2530*   2540
NG      002550          1587    1589    1592*
NXT1    017426          3073    3088    3091    3104*
NXT2    017522          3106    3112    3119    3122    3138*
OSPFLG  001326          1446*   1531*   2374    2393*
PASTBL  001246          1408*   2241    2256
PATTRN  001362          1463*   1963*   1967    1974*   1977*   2091*   2111*   2114*   2187    2911*   2918    2949*   2951
```

```
PC     =%000007          1257#  1633*  1637*  1705*  1706*  1707*  1717*  1719*  1767*  1769*  1784*  1785*  1795*
                         1787*  1809*  1813*  1815*  1928*  1830*  1836*  1849*  1870*  1871*  1894*  1895*  1902*
                         1935*  1953*  1964*  1966*  1970*  1979*  1981*  2001*  2022*  2029*  2041*  2044*  2049*
                         2057*  2066*  2092*  2093*  2102*  2103*  2116*  2121*  2130*  2142*  2172*  2176*  2232*
                         2303*  2304*  2343*  2349*  2433*  2450*  2805*  2811*  2817*  2820*  2823*  2849*  2865*
                         2914*  2921*  2928*  2942*  2954*  2964*  2977*  3014*  3048*  3290*  3612*  3619*  3626*
                         3640*  3642*
PIRQ   = 177772          1243#
PIRQVE= 000240           1337#
PRONUM  001312           1435#  1665*  1690*
PRO1    003246           1659   1690#
PRO2    003120           1657   1665#  1689
PRTTWO  014546           2824#
PR0    = 000000          1260#
PR1    = 000040          1261#
PR2    = 000100          1262#
PR3    = 000140          1263#
PR4    = 000200          1264#
PR5    = 000240          1265#
PR6    = 000300          1266#
PR7    = 000340          1267#
PS     = 177776          1240#  1241
PSW    = 177776          1241#
PWRVEC= 000024           1332#  1506*  1507*  3905*  3906*  3915*  3921*  3933*  3934*
RBA     001414           1478#  2574   2602   2603
RBUFF   013622           1478   2749#
RDBUFF  007316           2098   2186   2308#  2925   2931   2974   2981   3446
RDCHK   006410           2103   2181#
RDCHKD  016642           2968   2973#
RDCHR  = 104405          1581   1653   1671   2767   3770   3898#
RDLIN  = 104406          1623   3379   3838   3899#
RDLINK  005572           1707   1718   1787   2073#  2304
RDOCT  = 104407          1753   2294   3900#
RDTBL   001226           1407#
RD1     005616           2079#  2084   2095   2118
RD2     005630           2078   2082#
RD3     005636           2083   2085#
RD4     005700           2094   2096#  2117
RD5     005706           2097#  2106   2112
RD6     005712           2098#
RD7     006012           2109   2113#
READCS  001356           1461#  2101   2927   2976
RESTRT  006172           2136   2149#
RESVEC= 000010           1327#
RETFR2  007230           2289#
RETRN2  006406           2139   2148   2176#
RETRN3  005334           2026   2028#
RETRN4  003776           1804   1806#
RETRY   013100           2625   2634#  2644   2654   2729
RFCERR  013146           2605   2649#
RFERR   013130           2583   2642#
RKBA    001374           1468#  2038*  2557*  2574*  2591*  2918*  2925*  2951*  2974*  3446*  3499
RKCS    001370           1466#  1907*  1909   1934*  1936   2040*  2042   2126   2129*  2132   2140*  2143   2342*
                         2344   2348*  2350   2399*  2400   2432*  2434   2449*  2451   2517   2519*  2522   2524*
                         2558*  2559*  2560*  2561*  2562*  2563   2565   2575*  2576*  2577*  2578*  2579*  2580
                         2582   2592*  2593*  2594*  2595*  2596*  2612   2614   2634*  2635*  2650   2720   2819*
```

```
                    2821    2849*   2850    2858    2864*   2866    2874    2913*   2915    2920*   2922    2927*   2929
                    2953*   2955    2963*   2965    2976*   2978    3028*   3029    3047*   3085*   3086    3146*   3148
                    3234*   3235    3238*   3240    3257*   3258    3275*   3277    3408*   3409    3411*   3412    3425
                    3427*   3428    3435*   3436    3447*   3449    3492*   3493    3500
RKDA    001376      1469#   1906*   1917*   1933*   2037*   2333*   2341    2347*   2430*   2431*   2440*   2447*   2448*
                    2458*   2515*   2516*   2555*   2556*   2572*   2573*   2589*   2590*   2655*   2657*   2668    2818*
                    2847*   2912*   2917*   2924*   2950*   2973*   3036*   3067*   3068*   3144*   3233    3237*   3271*
                    3274*   3407*   3434*   3444*   3482*   3497    3548*
RKDS    001364      1464#   1918    1925    1938    2137    2145    2334    2394    2441    2459    2520    2525    2636
                    2943    3037    3049    3069    3079    3104    3108    3140    3151    3187    3196    3242    3272
                    3281    3483    3520    3543
RKER    001366      1465#   2852    2868    2957    2967
RKWC    001372      1467#   2039*   2554*   2571*   2588*   2919*   2925*   2952*   2975*   3445*   3498
RWC     001416      1479#   2511*   2571    2600
RWS   = 000100      1480#   2441    2459    3049    3187    3243    3281    3520
RWSRDY  011302      2375    2394#   2395
R0    =%000000      1248#   1582*   1586*   1588    1590*   1591    1624*   1628    1635    1694*   1698*   1699    1701
                    1704    1709*   1710    1712    1714    1716    1719    1754*   1755    1756*   1757*   1759*   1761*
                    1768    1776*   1782    1792    1843    1851    1900    1901*   1902*   1906    1917    1933    1952*
                    2073    2085*   2086    2087*   2088*   2089    2090*   2091    2120*   2170*   2171*   2173*   2243*
                    2245    2286    2305*   2331*   2336*   2339*   2346*   2347    2352    2366*   2370*   2373*   2429*
                    2431    2437    2439*   2440    2443    2446*   2448    2454    2457*   2458    2461    2494*   2497
                    2499*   2500*   2501*   2502*   2503*   2504*   2505    2768*   2772*   2774    2776    2777*   2779*
                    2779*   2780*   2781*   2782*   2783    3031*   3035*   3044*   3052    3064*   3067    3072    3074*
                    3090    3096*   3111    3117*   3121    3129*   3138    3154    3161*   3165    3172*   3214    3216*
                    3225    3227*   3271    3294    3381*   3382*   3384    3387*   3388*   3389*   3390*   3391    3392*
                    3394    3400*   3406*   3407    3434    3438*   3441*   3442*   3443*   3444    3490*   3492    3496*
                    3501    3513    3548    3595    3596*   3597    3600*   3835    3839*   3842    3858*   3869    3870*
                    3871    3872*   3873*   3874*   3875*   3907    3932*
R1    =%000001      1249#   1625*   1638    1640*   1716*   1755*   1758*   1763*   1765*   1766    1782*   1793    1823*
                    1826    1843*   1844    1858*   1860    1961*   1962*   1963    2076*   2079*   2080    2092    2095
                    2244*   2245*   2246*   2247*   2249*   2250    2252*   2253    2255    2301*   2332*   2333    2336
                    2337*   2367*   2370    2371*   2403*   2404*   2405    2412*   2413*   2414*   2415*   2416*   2417*
                    2420    2422    3032*   3041    3053*   3065*   3076    3093    3115    3126    3153    3169    3211
                    3218    3229    3285*   3286    3380*   3381    3392    3396    3416*   3419*   3421*   3424*   3443
                    3497*   3501*   3516*   3528*   3563    3836    3840*   3844*   3846*   3848*   3851*   3954    3957*
                    3908    3931*
R2    =%000002      1250#   1738*   1766*   1768*   1772*   1775*   1776    1777    1779    1780*   1783*   1803    1805*
                    1810    1814*   1848*   1853    1855    1960*   1961*   1862    1863*   1864*   1865*   1866*   1867*
                    1868    1869    1978*   1988*   1993    2007*   2008    2036*   2115*   2239*   2240*   2241    2250*
                    2255*   2256*   2278    2282    2284    2295*   2296*   2341*   3033*   3035    3036    3054*   3430*
                    3431*   3432    3498*   3504*   3517*   3529*   3552*   3837    3941*   3845*   3847*   3849*   3955
                    3856*   3909    3930*
R3    =%000003      1251#   1737*   1745    1770*   1847*   1848    1850*   1851    1853*   1855*   1856    1990*   1933
                    1997*   2006*   2013    2023*   2025    2237*   2265*   2270    2425*   2463*   2980*   2985*   3499*
                    3503*   3518*   3530*   3551*   3677    3686*   3692*   3693*   3696*   3701*   3702*   3703    3712*
                    3765    3767*   3769    3771*   3772    3779*   3780    3782    3790    3794    3796*   3802    3804
                    3806*   3809*   3910    3929*
R4    =%000004      1252#   1736*   1772    1773*   1805    1806*   1991*   1995    1996*   2008*   2013    2014*   2015*
                    2016*   2017*   2018*   2019*   2021    2181    2186*   2188    2224    2238    2231*   2397*   2398
                    2300    2301    2364*   2368*   2426*   2430    2931*   2933    2940    2981*   2983    2991    2996
                    3010    3012    3194*   3200    3202*   3500*   3505*   3506    3519*   3531*   3532    3534    3545*
                    3546    3553*   3554    3557    3678    3680*   3681*   3682*   3683    3684*   3698    3700*   3708*
                    3711*   3911    3928*
R5    =%000005      1253#   1680*   1681*   1682*   1987    1992*   1995*   1999*   2000*   2077*   2097    2105    2107*
                    2182    2187*   2188    2218    2230*   2402*   2403    2418    2427*   2447    2932*   2933    2992*
```

```
                    2983    3004    3185*   3189*   3195*   3198*   3233*   3237    3242*   3245*   3495*   3502    3509*
                    3510    3513*   3514*   3516    3528    3549*   3550    3565*   3566    3679    3685*   3697*   3699*
                    3690*   3691*   3692    3710*   3912    3927*
R6      =%000006    1254*   1499*   1499*   1500
R7      =%000007    1255*
SEC     013444      2672*   2676*   2698*
SECCNT  001344      1456*   2100    2919    2926    2952    2975
SECOND  003412      1526    1731*
SECONE  006712      1846    2237*
SECSYS  003010      1639    1647*   1664    1829
SECTBL  001305      1428*   2077    2107
SECT.0  013712      1602    2762*
SECT.1  011712      1601    2479*
SECT.2  010330      1600    2325*
SECT.3  002602      1599    1613*
SECT.4  017122      1603    3027*
SECT.5  020456      1604    3364*
SECT.6  021350      1605    3480*   3488
SEEKI   001402      1471*   2420*   2427
SEEKO   001404      1472*   2422*   2426
SEKSET  011500      2421    2425*
SHFCNT  020266      2485    2487    2489*   3294*
SHF1    013502      2677    2681    2714*
SHF2    013500      2713*
SHF3    013476      2673    2685    2689    2712*
SIN     = 001000    1492*   3196
SINST   020022      3197    3225*
SMTME   005554      1908    1935    2041    2063*   2130    2142    2343    2349    2433    2450    2820    2949    2965
                    2914    2921    2928    2954    2964    2977    3048    3290
SP      =%000006    1256*   1502*   1510*   1518*   1522    1582    1624    1654    1672    1745*   1754    1792*   1809*
                    1812*   1829*   1831*   1881*   1900*   1952    1987*   2000    2012*   2013*   2014    2028*   2073*
                    2120    2169*   2170    2174*   2181*   2182*   2197*   2210*   2218*   2224*   2230    2231    2270*
                    2278*   2295    2497*   2768    2788*   2996*   3004*   3010*   3041*   3076*   3093*   3115*   3136*
                    3158*   3169*   3211*   3219*   3229*   3380    3490*   3563*   3595*   3596    3597*   3599    3600
                    3601*   3603    3605    3607    3613    3615*   3617*   3625*   3629    3633    3634    3638    3669*
                    3670    3671    3672*   3677*   3678*   3679*   3685    3710    3711    3712    3713*   3714*   3737*
                    3738*   3741*   3742*   3743    3747*   3749*   3749    3752    3754    3756*   3765*   3766*   3771
                    3774    3778*   3785    3789*   3808    3809    3810*   3811*   3812*   3833*   3834*   3835*   3836*
                    3837*   3839    3842*   3850*   3851    3853    3854*   3856    3857    3858    3869*   3870    3890*
                    3881*   3907*   3908*   3909*   3910*   3911*   3912*   3913*   3914    3922*   3926    3927    3928
                    3929    3930    3931    3932
STACK = 001100      1231*   1502    2174
START   001440      1227    1495*   1726    2547    2633    2708    2910
STARTR  001434      1220    1494*
STFLG   001325      1445*   1527    1530*
STKLMT= 177774      1242*
STOP    013470      2707*
STORE   004032      1633    1819*
STRT1   001626      1531*   1598
SUR     013456      2680*   2701*
SWR     001140      1365*   1500    1512*   1514    1520*   2364    2402    2487    3401    3430    3439    3452    3913
                    3926*
SWREG   000176      1218*   1520
SWTCH   021334      3401*   3402*   3417    3422    3432    3457*
SWC     = 000001    1295*
SWCC    = 000001    1285*   1295
```

```
SW01   = 000002          1284*   1294
SW02   = 000004          1283*   1293
SW03   = 000010          1282*   1292
SW04   = 000020          1281*   1291
SW05   = 000040          1280*   1290
SW06   = 000100          1279*   1289
SW07   = 000200          1278*   1288
SW08   = 000400          1277*   1287
SW09   = 001000          1276*   1286
SW1    = 000002          1294*
SW10   = 002000          1275*
SW11   = 004000          1274*
SW12   = 010000          1273*
SW13   = 020000          1272*
SW14   = 040000          1271*
SW15   = 100000          1270*
SW2    = 000004          1293*
SW3    = 000010          1292*
SW4    = 000020          1291*
SW5    = 000040          1290*
SW6    = 000100          1289*
SW7    = 000200          1288*
SW8    = 000400          1287*
SW9    = 001000          1286*
SYSER    013046          2626*   2645    2730
SYSERR   013560          2728    2730*
S10      012070          2488    2494*
S11      012046          2489*   2546
S12      012166          2511*
S13      012036          2487*   2493
TABLTY   002476          1529    1576*   1597
TBITVE=  000014          1328*
TBLFUL   004074          1827    1829*
TIME     005440          2043    2050*
TIMR     001346          1457*   1903*   1911*   1932*   1940*   2034*   2050*   2063*   2064*
TIMR2    001323          1443*   1904*   1942    1944*   2036*   2053    2055*
TKVEC =  000060          1335*
TPVEC =  000064          1336*
TRAPVE=  000034          1334*   1504*   1505*
TRTVEC=  000014          1329*
TRYAGN   011312          2397*   2411
TSTSN1   006114          2128    2135    2137*
TYPE   = 104401          1536    1540    1544    1548    1552    1556    1560    1564    1568    1572    1577    1593    1615
                         1619    1648    1655    1660    1667    1673    1685    1722    1732    1740    1749    1788    1832
                         1877    1885    1889    1913    1920    1927    1947    2058    2152    2159    2163    2192    2201
                         2206    2214    2220    2257    2261    2266    2274    2290    2317    2326    2355    2359    2376
                         2380    2384    2388    2407    2471    2479    2495    2496    2756    2763    2784    2792    2796
                         2800    2806    2812    2825    2829    2833    2838    2842    2854    2860    2870    2876    2991
                         2885    2889    2894    2898    2902    2906    2935    2945    2959    2969    2987    2992    3000
                         3006    3040    3043    3057    3075    3078    3081    3092    3095    3114    3119    3125    3138
                         3157    3160    3168    3171    3207    3217    3220    3228    3231    3248    3365    3369    3373
                         3377    3489    3522    3539    3559    3608    3704    3777    3783    3788    3792    3797    3798
                         3800    3803    3807    3892*   3935
TYPEIT   013564          2541    2626    2658    2693    2734*   2735    2739
TYPOC  = 104402          2498    3042    3077    3094    3116    3127    3159    3170    3212    3219    3230    3491    3564
                         3893*
```

```
TYPON = 104404          2219    2225    3005    3011    3895#
TYPOS = 104403          1583    1746    1793    1882    2198    2211    2271    2279    2769    2789    2997    3894#
WATSK   017652          3166    3185#
WC      001412          1477#   2512*   2554    2588
WCERR   013510          2619    2720#
WCERRR  013026          2615    2619#
WCHERR  013540          2721    2726#
WFERR   013032          2566    2623#
WPS   = 000040          1491#   3079    3108
WRDCNT  001340          1454#   1968*   2039    2100*
WRITCS  001354          1460#   1969    2920    2953
WR_INK  005044          1706    1786    1959#
WRPRO   016312          2811    2823    2943#
WRRDCK  016102          2805    2817    2911#   2939
WRTNBY  001317          1439#   2089*   2197
X       015440          2875    2880#
XX      015654          2693#
$AJTOB  001134          1362#
$BDADR  001122          1357#
$BDDAT  001126          1359#
$CHARC  022224          3610*   3620*   3627    3636*   3641#
$CKSWR= ****** U        3898
$CMTAG  001100          1345#   1497    1498
$CM3  = 000000          1375#
$CNTLG  023067          3818#
$CNTLU  023062          3792    3817#
$CRLF   001161          1376#   2495    3057    3609    3644    3797    3817
$ENDAD  001400          1470#
$ERFLG  001103          1348#
$ERMAX  001115          1354#
$ERRPC  001116          1355#
$ERRTB  001164          1393#
$ERTTL  001112          1352#
$FILLC  001156          1373#   3613    3644
$FILLS  001155          1372#   3644
$GDADR  001120          1356#
$GDDAT  001124          1358#
$GTSWR= ****** U        3897
$HD   = 000003          1203    1204
$HIOCT  023216          3855*   3860#
$ICNT   001104          1349#
$ILLUP  023434          3905    3921    3938#
$INTAG  001135          1363#
$ITEMB  001114          1353#
$LF     001162          1377#   3644    3807    3817
$LPADR  001106          1350#
$LPERR  001110          1351#
$MAIL = ****** U        1524    3597
$MNEW   023105          3821#
$MSWR   023074          3819#
$NULL   001154          1371#   3615    3644
$OCNT   022452          3676#   3705*   3718#
$OMODE  022454          3671*   3675*   3680    3683*   3694*   3720#
$PASS   001100          1346#   2300*   2305
$POWER  023442          3936    3941#
$PWRDN  023274          1506    3905#   3933
```

```
$PWRMG  023430          3936#
$PWRUP  023346          3915    3921#
$QUES   001160          1375#   3644    3800    3817
$RDCHR  022456          3737#   3898
$RDDEC= ****** U        3901
$RDLIN  022576          3765#   3899
$RDOCT  023116          3833#   3900
$RDSZ = 000030          3758#
$R2A  = ****** U        3901
$SAVRE= ****** U        3901
$SAVR6  023440          3914#   3922    3923#   3924#   3940#
$SETUP= 000014          1494#   1503    1504    1506    1508    1536    1537    3726    3823
$STUP = 177777          1494#
$SWR  = 160000          1203    1204#   1375    3937
 TKB    001146          1368#   3724    3741    3747
 TKS    001144          1367#   3724    3739    3745
 TN   = 000001          1203#
 TPB    001152          1370#   2738#   3633#   3644
$TPFLG  001157          1374#   3591    3644
$TPS    001150          1369#   2734    3631    3644
$TRAP   023220          1504    3869#
$TRAP2  023242          3880#   3891
$TRP  = 000010          3884#   3893#   3894#   3895#   3896#   3898    3899#   3900#   3901#
$TRPAD  023254          3874    3891#
$TSTNM  001102          1347#
$TTYIN  023032          3767    3768    3780    3798    3812    3816#
$TYPBN= ****** U        3896
$TYPDS= ****** U        3896
$TYPE   022010          3591#   3884    3892
$TYPEC  022160          3612    3619    3626    3631#   3632
$TYPEX  022226          3637    3639    3642#
$TYPOC  022254          3674#   3893
$TYPON  022270          3673    3676#   3895
$TYPOS  022230          3669#   3894
$DFILL  022453          3670#   3674#   3684    3719#
.     = 023452          1212#   1216#   1225#   1344#   1378    1396#   1407#   1408#   1426#   1448#   1501    1543#   1547#
                        1551#   1555#   1559#   1563#   1567#   1571#   1575#   1618#   1622#   1735#   1743#   1791#   1889#
                        1916#   2061#   2204#   2223#   2269#   2308#   2329#   2401    2410#   2482#   2545#   2667#   2742
                        2749#   2787#   2803#   2828#   2836#   2841#   2845#   2892#   2901#   2939#   2948#   2962#   2995#
                        3009#   3084#   3210#   3334#   3376#   3437    3478#   3494    3507    3525#   3533    3542#   3544
                        3547    3555    3562#   3644    3724    3816#   3817    3823    3917    3939
```

DZRKIC.SRC CROSS REFERENCE TABLE -- MACRO NAMES

```
COMMEN      1#   1338#
ENDCOM      1#   1339#
ERROR    1232#
ESCAPE      1#   1338#
GETPRI      1#   1338#
GETSWR      1#   1192#   1338#   1537#
MULT        1#   1338#
NEWTST      1#   1338#
POP         1#   1338#   3856    3926    3927
PUSH        1#   1338#   3835    3907    3913
REPORT      1#   1339#
SCOPE    1233#
SETPRI      1#   1338#
SETTRA   3884#   3893    3894    3895    3898    3899    3900
SETUP       1#   1338#   1495
SKIP        1#   1338#
SLASH       1#   1338#
SPACE    1338#
STARS       1#   1338#   1340    1378    3576    3646    3723    3729    3758    3825    3863    3903    3919
SWRSU       1#   1338#   1508#
TRMTRP   3884#
TYPBIN      1#   1338#
TYPDEC      1#   1339#
TYPNAM      1#   1192#   1338#   1532
TYPNUM      1#   1338#
TYPOCS      1#   1338#   1881    2788
TYPOCT      1#   1338#
TYPTXT      1#   1338#   1540    1544    1548    1552    1556    1560    1564    1568    1572    1576    1592    1614    1619
         1647    1660    1667    1684    1721    1731    1739    1749    1798    1832    1877    1885    1889    1913    1920
         1927    1946    2058    2152    2157    2162    2192    2201    2205    2214    2220    2257    2261    2266    2274
         2289    2316    2325    2355    2359    2376    2380    2384    2388    2407    2470    2478    2755    2762    2784
         2792    2796    2800    2806    2812    2824    2829    2833    2838    2842    2854    2860    2870    2876    2880
         2885    2889    2894    2898    2902    2906    2935    2945    2959    2969    2987    2992    3000    3006    3091
         3206    3248    3365    3369    3373    3522    3539    3559
$$CMRE   1338#
$$CMTM   1338#
$$ESCA      1#   1338#
$$NEWT      1#   1338#
$$SET    3884#   3893    3894    3895    3898    3899    3900
$$SKIP      1#   1338#
.EQUAT      1#   1192#   1228
.HEADE      1#   1192#   1193
.KT11       1#
.SETUP      1#   1192#   1494
.SWRHI      1#
.$ACTI      1#
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#   1192#   1210
.$CMTA      1#   1192#   1338
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
```

```
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#   1192#   3901
.$RAND      1#
.$PDDE      1#
.$RDOC      1#   1192#   3823
.$READ      1#   1192#   3721
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#   1192#   3861
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#   1192#   3574
.$TYPO      1#   1192#   3644
.$4OOP      1#
.1170       1#
```

DZRKIC.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADD 1698 1814 2014 2028 2079 2169 2337 2371 2538 2602 2657 2671 2675 2679 2683
2687 2691 3054 3419 3424 3486 3601 3672 3682 3851
ASL 2489 3844 3846 3848 3873
ASR 2712 2713 2714
BCC 1760 1764 1774 1807 1998 2248 2369
BEQ 1528 1630 1636 1639 1657 1659 1700 1711 1779 1811 1827 1845 1926 1994 2011
2026 2046 2081 2128 2139 2146 2151 2189 2227 2285 2365 2395 2442 2460 2492
2521 2526 2637 2653 2724 2728 2737 2742 2869 2875 2934 2968 2984 2986 3050
3080 3112 3139 3152 3166 3215 3282 3287 3393 3405 3415 3418 3440 3453 3521
3604 3639 3699 3786 3843
BGT 3706 3755
BIC 1757 1758 1775 1855 1861 1902 1976 1988 2020 2088 2104 2113 2246 2296 2404
2417 2655 2656 2670 2674 2678 2682 2686 2690 3068 3096 3117 3172 3216 3274
3398 3402 3431 3438 3442 3596 3742 3748 3756 3850
BICB 2252
BIS 1679 1761 1765 1772 1853 1973 2021 2096 2110 2171 2173 2249 2431 2448 2516
2556 2559 2560 2561 2562 2573 2576 2577 2578 2579 2590 2593 2594 2595 2596
3044 3074 3129 3144 3161 3227 3406 3441 3443 3514 3701 3702
BISB 1665 2075 2097 2196
BIT 1712 1810 1844 1925 1938 1971 1993 2108 2126 2137 2145 2282 2284 2394 2418
2441 2459 2487 2520 2525 2582 2636 2720 2741 2852 2858 2868 2874 2943 2957
2967 3049 3079 3108 3111 3121 3138 3151 3154 3165 3187 3196 3214 3225 3243
3281 3439 3452 3520
BITB 3627
BLE 1589 1677 1821 2775
BLO 3781
BLOS 3769
BLT 3383 3385 3618 3707 3753
BMI 1587 1632 1675 1715 1804 1910 1919 2251 2299 2566 2615 2773 3073 3141 3484
BNE 1501 1515 1535 1713 1852 1857 1912 1939 1941 1943 1972 2052 2054 2065 2106
2109 2135 2148 2191 2229 2242 2283 2338 2353 2372 2375 2419 2438 2444 2455
2462 2464 2488 2528 2540 2583 2605 2607 2625 2644 2721 2853 2859 2941 2944
2958 3013 3055 3109 3122 3155 3198 3190 3197 3199 3201 3226 3244 3246 3395
3397 3403 3423 3433 3487 3511 3567 3598 3606 3614 3628 3635 3697 3744 3750
3773 3775 3791 3795 3805 3925
BPL 1525 1702 1937 2043 2083 2133 2144 2168 2335 2345 2351 2401 2406 2435 2452
2518 2523 2564 2581 2613 2651 2735 2822 2851 2867 2916 2923 2930 2956 2966
2979 3030 3038 3070 3087 3091 3106 3149 3236 3241 3259 3273 3278 3410 3413
3426 3429 3437 3450 3494 3507 3533 3536 3544 3547 3555 3558 3592 3632 3695
3740 3746
BR 1517 1537 1541 1545 1549 1553 1557 1561 1565 1569 1573 1578 1594 1597 1616
1620 1634 1641 1649 1661 1664 1668 1683 1686 1689 1696 1703 1708 1720 1723
1733 1741 1750 1771 1781 1789 1797 1833 1854 1878 1886 1890 1914 1921 1924
1928 1931 1945 1948 1951 1965 1975 1980 2027 2048 2056 2059 2062 2078 2084
2094 2095 2112 2117 2118 2153 2156 2159 2164 2193 2202 2207 2215 2221 2254
2258 2262 2267 2275 2291 2318 2327 2356 2360 2377 2381 2385 2389 2409 2411
2421 2423 2465 2472 2480 2493 2546 2645 2654 2739 2757 2764 2785 2793 2797
2801 2807 2813 2826 2830 2834 2839 2843 2855 2861 2871 2877 2882 2886 2890
2895 2899 2903 2907 2936 2939 2946 2960 2970 2988 2993 3001 3007 3082 3098
3119 3163 3203 3208 3213 3221 3249 3252 3366 3370 3374 3420 3489 3523 3527
3540 3560 3594 3611 3621 3630 3637 3673 3688 3709 3751 3784 3793 3799 3801
3852 3917 3939
CLC 1762 3386
CLR 1499 1627 1824 1903 1932 1991 2012 2034 2047 2237 2240 2332 2367 2486 2911
2932 2982 3032 3033 3065 3194 3242 3378 3480 3481 3495 3549 3686 3766 3789
3840 3841 3923

CLRB	1494	1531	1626	1690	1695	1904	1959	2036	2183	3610	3636	3796	3806		
CMP	1500	1514	1588	1638	1656	1658	1676	1699	1710	1778	1819	1826	1851	1856	2025
	2080	2105	2188	2228	2241	2352	2405	2437	2443	2454	2461	2491	2533	2604	2606
	2624	2643	2652	2723	2727	2774	2933	2940	2983	3012	3286	3284	3394	3432	3510
	3566	3743	3749	3752	3754	3768	3780								
CMPB	1629	1635	2010	3603	3605	3613	3634	3638	3772	3790	3794	3804			
COM	2536														
COMB	1974	1977	2111	2114											
DEC	2064	2226	2463	2985	3198	3779									
DECB	2134	2147	3617	3620	3694	3705									
EMT	1232														
HALT	1215	1796	1893	2287	2363	2397	2483	2632	2707	2804	2810	2916	2837	2846	2893
	3593	3916	3938												
INC	1534	1770	1911	1940	1978	2023	2050	2115	2265	2490	2527	2623	2642	2649	2722
	2726	2740	2743	3053	3189	3202	3245	3285	3399	3485	3509	3565	3700	3708	3924
INCB	1530	1825	1873	1944	1996	2055	2393	3640							
IOT	1233														
JMP	1220	1227	1526	1529	1591	1726	1846	2136	2175	2306	2547	2619	2633	2638	2708
	2725	2729	2730	2910	3142	3288	3289	3454	3569						
JSR	1633	1637	1705	1706	1707	1717	1718	1767	1769	1784	1785	1786	1787	1849	1870
	1894	1908	1935	1964	1970	1979	2041	2044	2057	2092	2093	2102	2103	2116	2130
	2142	2303	2304	2343	2349	2433	2450	2531	2533	2535	2537	2541	2626	2658	2673
	2677	2681	2685	2689	2693	2805	2811	2817	2820	2823	2849	2865	2914	2921	2928
	2954	2964	2977	3048	3280	3612	3619	3626							
MOV	1498	1502	1504	1505	1506	1507	1510	1511	1512	1513	1518	1520	1521	1522	1582
	1624	1625	1640	1654	1672	1682	1694	1704	1709	1716	1736	1737	1738	1745	1754
	1776	1780	1782	1792	1808	1812	1823	1829	1831	1843	1850	1881	1900	1901	1906
	1907	1917	1933	1934	1952	1963	1967	1968	1969	1987	1990	1992	2000	2006	2007
	2037	2038	2039	2040	2063	2073	2076	2077	2085	2086	2091	2098	2100	2101	2107
	2120	2129	2131	2140	2141	2170	2174	2181	2182	2185	2186	2187	2210	2218	2224
	2230	2231	2239	2243	2244	2256	2270	2278	2295	2297	2300	2301	2305	2331	2333
	2336	2339	2341	2342	2346	2347	2348	2366	2370	2373	2399	2402	2420	2422	2425
	2426	2427	2429	2430	2432	2439	2440	2446	2447	2449	2457	2458	2485	2494	2497
	2505	2506	2507	2508	2509	2510	2511	2512	2513	2514	2515	2519	2524	2529	2530
	2532	2534	2554	2555	2557	2558	2567	2571	2572	2574	2575	2584	2588	2589	2591
	2592	2600	2603	2608	2616	2617	2634	2635	2668	2669	2715	2768	2783	2788	2818
	2819	2847	2848	2864	2912	2913	2917	2918	2919	2920	2924	2925	2926	2927	2931
	2949	2950	2951	2952	2953	2963	2973	2974	2975	2976	2980	2981	2996	3004	3010
	3028	3031	3035	3036	3041	3047	3064	3067	3076	3085	3093	3115	3126	3146	3158
	3169	3185	3195	3211	3218	3229	3233	3234	3237	3238	3257	3271	3275	3380	3391
	3400	3401	3407	3408	3411	3416	3421	3427	3430	3434	3435	3444	3445	3446	3447
	3482	3490	3492	3497	3498	3499	3500	3501	3502	3503	3504	3505	3513	3515	3516
	3517	3518	3519	3528	3529	3530	3531	3545	3548	3550	3551	3552	3553	3563	3595
	3596	3600	3615	3669	3677	3678	3679	3685	3692	3710	3711	3712	3713	3714	3737
	3738	3765	3767	3778	3809	3810	3811	3812	3833	3834	3835	3836	3837	3839	3854
	3855	3856	3857	3858	3869	3870	3874	3880	3881	3905	3906	3907	3908	3909	3910
	3911	3912	3913	3914	3915	3921	3922	3926	3927	3928	3929	3930	3931	3932	3933
	3934														
MOVB	1226	1628	1680	1755	1766	1768	1783	1805	1847	1848	1860	1862	1868	1869	1975
	1876	1961	1995	1999	2008	2013	2089	2197	2253	2255	2364	2403	2672	2676	2680
	2684	2688	2692	2738	2776	3381	3392	3597	3625	3633	3670	3671	3674	3675	3676
	3680	3683	3684	3703	3741	3747	3771	3776	3782	3787	3802	3842	3872		
NEG	2601	3681													
NOP	1613	1652	1666	1744	1820	1859	1874	1905	1960	1989	2009	2024	2035	2051	2074
	2099	2127	2138	2150	2184	2238	2302	2310	3027	3364					
RO_	1590	1681	1863	1864	1865	1866	1867	1962	2015	2016	2017	2018	2019	2090	2412

```
                2413  2414  2415  2416  2500  2501  2502  2503  2504  2779  2779  2780  2781  2782  3687
                3689  3690  3691  3693  3845  3847  3849
ROLB            2247
ROR             1759  1763  1773  1806  1997  2368  3387  3388  3389  3390
RTI             1519  3602  3715  3757  3813  3859  3882  3937
RTS             1809  1813  1815  1828  1830  1836  1871  1895  1953  1966  1981  2001  2022  2029  2049
                2066  2121  2172  2176  2232  2618  2716  2744  2942  3014  3642  3875
SUB             1586  1631  1674  2772  3382
SWAB            1678  1756  1822  1858  2087  2245  2499  2777
TRAP            3884  3893  3894  3895  3898  3899  3900
TST             1701  1714  1719  1777  1803  2045  2082  2286  2298  2565  2614  2991  3052  3200  3284
                3404  3414  3417  3422  3425  3534  3557  3599  3607  3629  3698  3774  3785  3808  3853
                3871
TSTB            1524  1527  1909  1918  1936  1942  2042  2053  2132  2143  2149  2167  2190  2250  2334
                2344  2350  2374  2400  2434  2451  2517  2522  2563  2580  2612  2650  2734  2736  2821
                2850  2866  2915  2922  2929  2955  2965  2978  3029  3037  3069  3072  3086  3090  3104
                3140  3148  3235  3240  3258  3272  3277  3396  3409  3412  3428  3436  3449  3483  3493
                3506  3532  3543  3546  3554  3591  3631  3739  3745
.ASCII          1375  1376  2695  2696  2697  2698  2699  2701
.ASCIZ          1377  1539  1543  1547  1551  1555  1559  1563  1567  1571  1575  1580  1596  1618  1622
                1651  1663  1670  1688  1725  1735  1743  1752  1791  1835  1880  1888  1892  1916  1923
                1930  1950  2061  2155  2161  2166  2195  2204  2209  2217  2223  2260  2264  2269  2277
                2293  2320  2329  2358  2362  2379  2383  2387  2391  2410  2474  2482  2543  2628  2662
                2705  2759  2766  2787  2795  2799  2803  2809  2815  2828  2832  2836  2841  2845  2857
                2963  2873  2879  2884  2888  2892  2897  2901  2905  2909  2938  2948  2962  2972  2990
                2995  3003  3009  3084  3210  3251  3297  3299  3301  3303  3305  3308  3311  3313  3315
                3368  3372  3376  3458  3525  3542  3562  3817  3818  3819  3821  3941
.BLKB            1426  3816
.BLKW           1396  1407  1408  2308  2749
.BYTE           1347  1348  1353  1354  1362  1363  1371  1372  1373  1374  1410  1411  1412  1413  1414
                1415  1416  1417  1419  1420  1421  1422  1423  1424  1428  1429  1430  1431  1433  1434
                1435  1436  1437  1438  1439  1440  1441  1442  1443  1444  1445  1446  1584  1585  1747
                1748  1794  1795  1883  1884  2199  2200  2212  2213  2272  2273  2280  2281  2770  2771
                2790  2791  2998  2999  3716  3717  3718  3719  3814  3815
.DSABL          3726
.ENABL             1  1192  3724
.END            3944
.ENDC           1198  1221  1232  1324  1338  1341  1345  1347  1375  1379  1494  1502  1503  1504  1506
                1508  1524  1536  1537  1539  1543  1547  1551  1555  1559  1563  1567  1571  1575  1580
                1596  1618  1622  1651  1663  1670  1688  1725  1735  1743  1752  1791  1835  1880  1984
                1885  1888  1892  1916  1923  1930  1950  2061  2155  2161  2166  2195  2204  2209  2217
                2223  2260  2264  2269  2277  2293  2320  2329  2358  2362  2379  2383  2387  2391  2410
                2474  2482  2759  2766  2787  2791  2792  2795  2799  2803  2809  2815  2828  2832  2836
                2841  2845  2857  2863  2873  2879  2884  2888  2892  2897  2901  2905  2909  2938  2948
                2962  2972  2990  2995  3003  3009  3084  3210  3251  3368  3372  3376  3525  3542  3562
                3577  3597  3647  3724  3725  3726  3730  3758  3759  3767  3769  3772  3800  3817  3823
                3826  3828  3861  3864  3870  3873  3892  3893  3894  3895  3896  3897  3898  3899  3900
                3901  3904  3913  3914  3920  3926  3927  3937  3944
.EQUIV          1232  1233  1241  1286  1287  1288  1289  1290  1291  1292  1293  1294  1295  1314  1315
                1316  1317  1318  1319  1320  1321  1322  1323
.EVEN           1448  1539  1543  1547  1551  1555  1559  1563  1567  1571  1575  1580  1596  1618  1622
                1651  1663  1670  1688  1725  1735  1743  1752  1791  1835  1880  1888  1892  1916  1923
                1930  1950  2061  2155  2161  2166  2195  2204  2209  2217  2223  2260  2264  2269  2277
                2293  2320  2329  2358  2362  2379  2383  2387  2391  2410  2474  2482  2545  2631  2667
                2706  2759  2766  2787  2795  2799  2803  2809  2815  2828  2832  2836  2841  2845  2957
                2863  2873  2879  2884  2888  2892  2897  2901  2905  2909  2938  2948  2962  2972  2990
```

```
        2995  3003  3009  3084  3210  3251  3334  3368  3372  3376  3478  3525  3542  3562  3943
.IF     1194  1219  1230  1296  1324  1340  1344  1346  1375  1378  1379  1494  1497  1502  1504
        1506  1508  1524  1535  1536  1537  1538  1542  1546  1550  1554  1558  1562  1566  1570
        1574  1579  1595  1617  1621  1650  1662  1669  1687  1724  1734  1742  1751  1790  1834
        1879  1883  1884  1887  1891  1915  1922  1929  1949  2060  2154  2160  2165  2194  2203
        2208  2216  2222  2259  2263  2268  2276  2292  2319  2328  2357  2361  2378  2382  2386
        2390  2409  2473  2481  2758  2765  2786  2790  2791  2794  2798  2802  2808  2814  2827
        2831  2835  2840  2844  2856  2862  2872  2878  2883  2887  2891  2896  2900  2904  2908
        2937  2947  2961  2971  2989  2994  3002  3008  3083  3209  3250  3367  3371  3375  3524
        3541  3561  3576  3597  3646  3723  3725  3726  3729  3730  3758  3766  3768  3772  3773
        3816  3817  3823  3825  3828  3840  3863  3869  3873  3884  3893  3894  3895  3896  3897
        3898  3899  3900  3901  3903  3913  3914  3919  3926  3927  3935  3937  3941
.IFF    1230  1341  1344  1346  1375  1379  1502  1535  1536  1884  2791  3577  3647  3724  3726
        3730  3732  3737  3758  3759  3768  3800  3816  3826  3864  3870  3904  3920  3937
.IFT    1539  1543  1547  1551  1555  1559  1563  1567  1571  1575  1580  1596  1618  1622  1651
        1663  1670  1688  1725  1735  1743  1752  1791  1835  1880  1888  1892  1916  1923  1930
        1950  2061  2155  2161  2166  2195  2204  2209  2217  2223  2260  2264  2269  2277  2293
        2320  2329  2358  2362  2379  2383  2387  2391  2410  2474  2482  2759  2766  2787  2795
        2799  2803  2809  2815  2828  2832  2836  2841  2845  2857  2863  2873  2879  2884  2888
        2892  2897  2901  2905  2909  2938  2948  2962  2972  2990  2995  3003  3009  3094  3210
        3251  3368  3372  3376  3525  3542  3562  3732  3737  3844  3860  3861
.IFTF   1539  1543  1547  1551  1555  1559  1563  1567  1571  1575  1580  1596  1618  1622  1651
        1663  1670  1688  1725  1735  1743  1752  1791  1835  1880  1888  1892  1916  1923  1930
        1950  2061  2155  2161  2166  2195  2204  2209  2217  2223  2260  2264  2269  2277  2293
        2320  2329  2358  2362  2379  2383  2387  2391  2410  2474  2482  2759  2766  2787  2795
        2799  2803  2809  2815  2828  2832  2836  2841  2845  2857  2863  2873  2879  2884  2888
        2892  2897  2901  2905  2909  2938  2948  2962  2972  2990  2995  3003  3009  3084  3210
        3251  3368  3372  3376  3525  3542  3562  3726  3730  3733  3840  3844  3860
.IIF    1193  1198  1203  1204  1216  1378  1503  1536  1882  2789  3644  3724  3808  3817  3823
        3892  3893  3894  3895  3898  3899  3900
.IRP    1494  3835  3856  3907  3913  3926  3927
.LIST      1  1192  1216  1338  1375  1494  1508  1536  1537  1539  1543  1547  1551  1555  1559
        1563  1567  1571  1575  1580  1596  1618  1622  1651  1663  1670  1688  1725  1735  1743
        1752  1791  1835  1880  1888  1892  1916  1923  1930  1950  2061  2155  2161  2166  2195
        2204  2209  2217  2223  2260  2264  2269  2277  2293  2320  2329  2358  2362  2379  2383
        2387  2391  2410  2474  2482  2759  2766  2787  2795  2799  2803  2809  2815  2828  2932
        2836  2841  2845  2857  2863  2873  2879  2884  2888  2892  2897  2901  2905  2909  2939
        2948  2962  2972  2990  2995  3003  3009  3084  3210  3251  3368  3372  3376  3525  3542
        3562  3758  3884  3892  3893  3894  3895  3896  3898  3899  3900  3901
.MACRO     1  1338  3884
.MCALL  1192  1338  1508  1537
.NLIST     1  1192  1216  1338  1375  1494  1508  1536  1537  1539  1543  1547  1551  1555  1559
        1563  1567  1571  1575  1580  1596  1618  1622  1651  1663  1670  1688  1725  1735  1743
        1752  1791  1835  1880  1888  1892  1916  1923  1930  1950  2061  2155  2161  2166  2195
        2204  2209  2217  2223  2260  2264  2269  2277  2293  2320  2329  2358  2362  2379  2383
        2387  2391  2410  2474  2482  2759  2766  2787  2795  2799  2803  2809  2815  2828  2832
        2836  2841  2845  2857  2863  2873  2879  2884  2888  2892  2897  2901  2905  2909  2939
        2948  2962  2972  2990  2995  3003  3009  3084  3210  3251  3368  3372  3376  3525  3542
        3562  3758  3884  3892  3893  3894  3895  3896  3898  3899  3900  3901
.PAGE   1338  1379
.REM       1
.REPT   1216
.SBTTL  1210  1219  1228  1338  1379  1496  1532  1608  2323  2476  2760  3021  3345  3463  3574
        3644  3721  3823  3861  3884  3901
.TITLE  1193
.WORD   1216  1217  1218  1346  1349  1350  1351  1352  1355  1356  1357  1358  1359  1360  1361
```

```
MAINDEC-11-DZRKI-C      MACY11 27(732)  23-SEP-76  10:03  PAGE 95
DZRKIC.SRC      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

        1364    1365    1366    1394    1398    1399    1400    1401    1402    1403    1404    1405    1450    1451    1452
        1453    1454    1455    1456    1457    1458    1459    1460    1461    1462    1463    1464    1465    1466    1467
        1468    1469    1470    1471    1472    1476    1477    1479    1480    1481    1482    1483    1484    1485    2748
        3294    3295    3335    3336    3337    3338    3339    3340    3341    3342    3571    3641    3720    3860    3891
        3926


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DZRKIC.SEQ/CRF/SOL/PAGNUM/NL:TOC=SYSMAC.SML,DZRKIC.SRC
 RUN-TIME: 45 50 6 SECONDS
 RUN-TIME RATIO: 189/102=1.8
 CORE USED:  34K  (67 PAGES)
```

N07

Spooler runtime 14 Seconds, 435 KCS, 390 disk reads, 3 disk writes, 90 pages