RK06/RK07

UNIBUS DUAL PORT DRIVE

MD-11-DZR6G-A

DIAGNOSTIC

EP-DZR6G-A-DL

FICHE 1 OF 2

JAN 1978

digital

MADE IN USA

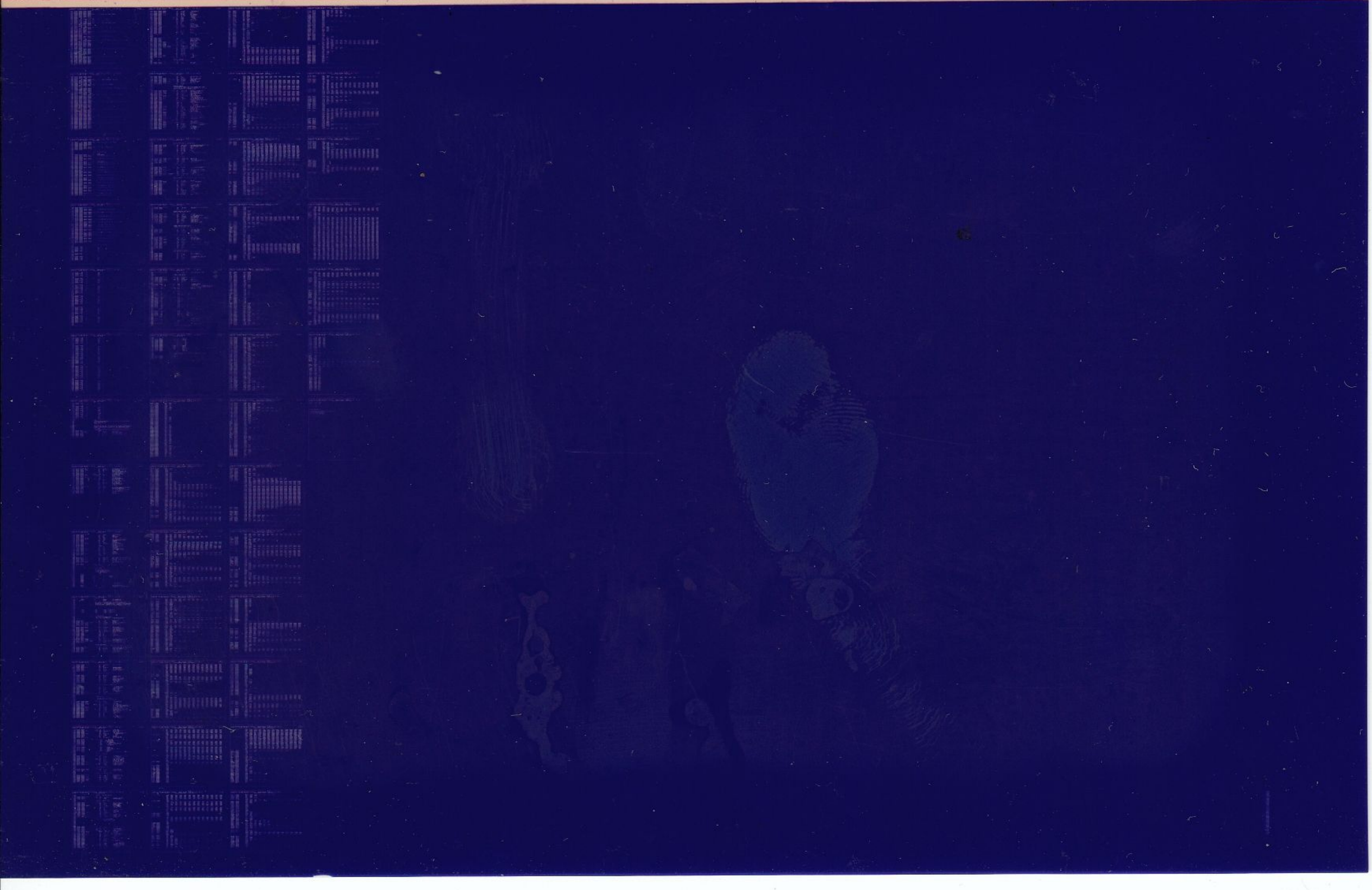
RK06 / RK07

UNIBUS DUAL PORT DRIVE
MD-11-DZR6G-A
DIAGNOSTIC

EP-DZR6G-A-DL

FICHE 2 OF 2

JAN 1978
digital
MADE IN USA



```
.REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6G-A-D
PRODUCT NAME:	UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC
DATE:	AUGUST 1977
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	GARY PAPAZIAN

TABLE OF CONTENTS

1.0 ABSTRACT

2.0 REQUIREMENTS

2.1 HARDWARE
2.2 PRELIMINARY TESTING & PROGRAMS
2.3 RESTRICTIONS & OPERATOR ACTIONS

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY
3.2 XXDP
3.3 ACT/APT
3.3.1 APT ETABLE DEFINITIONS
3.4 MEMORY MANAGEMENT
3.5 PARITY CHECK ENABLED
3.6 BAD SECTORS
3.7 EXECUTION TIME
3.8 FAULT ISOLATION
3.9 ERROR CORRECTION & FAILURE RATE ANALYSIS
3.10 DEFAULT UNIBUS ADDRESSES & VECTORS

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING
4.2 STARTING LOCATIONS
4.3 CONSOLE SWITCH REGISTERS
4.4 SOFTWARE SWITCH REGISTER
4.5 INPUT DIALOGUE
4.6 PROGRAM EXAMPLE
4.7 HALTING THE PROGRAM

5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL
5.2 TEST DESCRIPTIONS

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION
6.2 ERROR PRINTOUT EXAMPLE

7.0 DUAL PROCESSOR-DUAL CONTROLLER TESTING

1.0 ABSTRACT

THE RK06 DUAL PORT LOGIC PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE DUAL PORT OPTION IS FUNCTIONING PROPERLY.

BOTH PORTS OF THE RK06 ARE CABLED TO THE SAME RK611 BY A STANDARD CABLE & THE DUAL PORT TEST SWITCH IS ENABLED ON THE DUAL PORT MODULE. THIS ARRANGEMENT ALLOWS THE DUAL PORT, LOGIC TO BE TESTED FROM ONE PDP-11/RK611 TO A MAXIMUM OF 4 DRIVES.

THIS PROGRAM WILL TEST RK06/RK07 DRIVES WITHOUT OPERATOR INPUTS

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

- PDP-11
- CONSOLE TELETYPE
- 16K MEMORY
- KW11-L OR KW11-P CLOCK
- RK06 UNIBUS CONTROLLER (RK611)
- 1 TO 4 RK06/RK07 DRIVES
- FORMATTED DISKPACKS

2.2 PRELIMINARY TESTING & PROGRAMS

1. THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY ON BOTH PORTS.

2. THE RK06 DRIVE DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY.

3. THE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC SHOULD FIRST RUN SUCESSFULLY.

2.3 RESTRICTIONS & OPERATOR ACTION

TO TEST THE RK06 DUAL PORT OPTION WITH THIS PROGRAM, THE DUAL PORT TEST SWITCH MUST BE ENABLED ON THE DUAL PORT MODULE.

THE CABLE FROM THE RK611 IS DAISY CHAINED TO BOTH PORTS OF RK06 UNDER TEST. ENABLING THE SWITCH ON THE DUAL PORT MODULE, GROUNDS BIT 0 OF THE UNIT SELECT LINES GOING TO THE PORT B INTERFACE & TIMING MODULE (M7706).

THE EFFECT OF THIS IS THAT BIT 0 UNIT SELECT IS COMPLEMENTED

ON PORT b AND THE DRIVE APPEARS AS 2 SEPARATE UNITS TO THE
RK611. EACH PORT OF THE RK06 WILL RESPOND TO A DIFFERENT
DRIVE ADDRESS.

THE ADDRESS OF EACH PORT WILL DEPEND ON THE DRIVE ADDRESS. FOR
THIS REASON, THE RESTRICTION IS MADE THAT ONLY EVEN NUMBERED
UNIT SELECT PLUGS BE USED. IN THIS WAY PORT 'A' WILL RESPOND TO
THE DRIVE ADDRESS, & PORT 'B' WILL RESPOND TO ADDRESS+1 (THE ADD-
RESS DEVELOPED BY THE SWITCH).

BECAUSE OF THE ABOVE CONSIDERATIONS, A MAXIMUM OF 4 DRIVES CAN
BE TESTED BY THIS PROGRAM, WHICH WILL 'SEE' 8 DRIVES.

ANY OTHER DRIVES ON THE SYSTEM WHICH HAS ANY
ADDRESS IN CONFLICT WITH EITHER OF THE TEST ADDRESSES MUST BE
DESELECTED, BOTH PORTS SWITCHED OFF.

A FURTHER REQUIREMENT IS THE SYSTEM MUST HAVE EITHER A KW11-P
OR KW11-L. IF NEITHER IS PRESENT AN ERROR MESSAGE WILL BE TYPED
AND THE PROGRAM WILL JUMP TO THE END OF PASS.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

THIS PROGRAM CAN BE USED BY THE PDP-11/04,05,10,20,
34,35,40,45,50,55 & 70.

IT IS COMPATABLE WITH THE LSI-11 INSTRUCTION SET AND CAN TEST
THE RK06 ONLY IF THE DRIVE CONTROLLER FOR THE LSI-11 IS
DESIGNED TO BE DIAGNOSTICALLY COMPATABLE WITH THE RK611.

3.2 XXDP

THIS PROGRAM SHOULD NOT BE CHAINED BY XXDP.

CHAIN MODE OPERATION (MONITOR)

BY DEFINITION, ANY PROGRAM THAT REQUIRES OPERATOR INTERVENTION
SHOULD NOT BE CHAINED.

IN THIS CASE, OPERATOR INTERVENTION IS REQ'D TO ENABLE THE
DUAL PORT TEST SWITCH & DAISY CHAINING BOTH PORTS TO THE SAME
RK611.

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. IF THE LOADING MEDIUM ON DRIVE 0 IS AN RK06. IT WILL BE

```
                    TESTED.  THE OPERATOR IS FIRST GIVEN A MESSAGE TO REPLACE
                    THE PACK ON DRIVE 0 WITH A SCRATCH PACK & TYPE <CR>
                    WHEN DONE.


3.3     ACT/APT

THIS PROGRAM IS ACT COMPATIBLE.

HOWEVER, IT SHOULD BE RUN ONLY IN DUMP MODE.

AUTOMATIC MODE (MONITOR)

BY DEFINITION ANY PROGRAM THAT REQUIRES OPERATOR INTERVENTION
SHOULD NOT BE RUN IN THE AUTO MODE.

DUMP MODE (MANUAL):  INPUT DIALOGUE IF STARTED FROM 220.


3.3.1   APT ETABLE DEFINITIONS

THE FOLLOWING DEFINITIONS ARE VALID FOR SPECIFYING APT ENVIRONMENTAL
TABLE (ETABLE) ENTRIES.  VIA RUNNING THE APT UTILITY PROGRAM  "TSP":

1.  SOFTWARE ENVIRONMENT:
        =1 IF APT SCRIPT MODE
        =0 IF STANDALONE MODE

2.  ENVIRONMENT MODE: BYTE
        BIT 7 = 1  ETABLE DOES SIZING
              = 0  PROGRAM DOES SIZING
        BIT 6 = 1  SPOOL MESSAGES TO APT IF SCRIPT MODE
              = 0  DON'T SPOOL TO APT
        BIT 5 = 1  SUPPRESS CONSOLE OUTPUT
              = 0  ALLOW CONSOLE OUTPUT
        BITS 4-0   NOT USED

3.  SWITCH 1 (SOFTWARE SWITCH REGISTER)
    IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1, THE SOFTWARE
    SWITCH REGISTER WILL BE USED, INSTEAD OF THE HARDWARE CONSOLE
    SWITCH REGISTER.  REGARDLESS OF WHICH ONE IS USED, ALL BITS
    DEFINED IN SECTIONS 4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY USED
    WHEN RUNNING IN STANDALONE MODE.
    IN APT SCRIPT MODE, HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS

    BE SET TO 0.

4.  SWITCH 2 (USER SWITCH REGISTER)
    NOT USED

5.  CPU OPTIONS:
    NOT USED

6.  MEMORY TYPES 1-4 AND MAX MEMORY ADDRESSES
    NOT USED
```

```
                        7.  INTERRUPT VECTOR 1:
                            USED WHEN ENVIRONMENT MODE BIT 7=1.  DEFAULT = 210

                        8.  BUS PRIORITY 1:
                            USED WHEN ENVIRONMENT MODE BIT 7=1.  DEFAULT = 5

                        9.  INTERRUPT VECTOR 2:
                            NOT USED

                        10. BUS PRIORITY 2:
                            NOT USED

                        11. BASE ADDRESS:
                            USED WHEN ENVIRONMENT MODE BIT 7 = 1.  DEFAULT = 177440

                        12. DEVICE MAP:
                            USED WHEN ENVIRONMENT MODE BIT 7 = 1.  EACH BIT SET TO
                            1 IN BITS 0-7 WILL SELECT THE CORRESPONDING DRIVE TO BE
                            TESTED.  BITS 8-15 ARE NOT USED.

                        NOTE:   IN THIS PROGRAM, ONLY EVEN NUMBERED DRIVES CAN BE
                                TESTED (0,2,4,6)

                        13. CONTROLLER DESCRIPTORS:
                            NOT USED.

                        14. DRIVE DESCRIPTOR CODES (IN WORDS):
                            NOT USED


                3.4     MEMORY MANAGEMENT

                        MEMORY MANAGEMENT IS NOT USED.


                3.5     PARITY CHECK ENABLED


                        IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM,
                        THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.


                3.6     BAD SECTOR

                        THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR
                        INFORMATION CONTAINED ON CYLINDER 410, HEAD 2.  PRINTOUTS
                        OF DATA ERRORS DUE TO BAD SECTORS/TRACKS WILL BE MASKED OUT.


                3.7     EXECUTION TIME

                        THE EXECUTION TIME IS APPROX.  2.5 MIN  BASED ON THE PDP 11/50.


                3.8     FAULT ISOLATION
```

TO BE DETERMINED.

3.9 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OR FAILURE RATE ANALYSIS.

3.10 DEFAULT UNIBUS ADDRESSES & VECTORS

THE FOLLOWING IS A LIST OF ALL DEFAULT ADDRESSES & VECTORS OF ALL HARDWARE TO BE USED & THEIR MEMORY ADDRESSES WHERE THEY CAN BE CHANGED.

	LOCATION	DEFAULT CONTENTS
RK06-RK07 BUSS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1314	210
CONTROLLER PRIORITY	1316	240
P-CLOCK STATUS REG	1320	172540
P-CLOCK SET BUFFER	1322	172542
P-CLOCK READ BUFFER	1324	172544
L-CLOCK STATUS REG	1326	177546
L-CLOCK INTERRUPT VECTOR	1330	100
P-CLOCK INTERRUPT VECTOR	1332	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564
TTY PRINTER BUFFER	1152	177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING STANDARD PROCEDURE FOR ABSOLUTE LOADER TAPES; OR FROM ANY MEDIA SUPPORTED BY XXDP.

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

4.1.2 SET SWITCH REGISTERS AS DESIRED (SEE SEC 4.3).

4.1.3 INSTALL CALBE(S) & SET DRIVES TO BE TESTED IN THE 'LOAD' CONDITION & WITH BOTH PORTS SELECTED & WRITE LOCK DISABLED. DRIVES NOT TO BE TESTED MUST HAVE BOTH PORTS DESELECTED. ALSO, THE DUAL PORT TEST SWITCH ON THE DUAL PORT MODULE MUST BE ENABLED.

NOTE: THE DRIVE WILL NOT RESPOND TO THE 'START SPINDLE' COMMAND IF THE RUN/STOP SWITCH IS IN THE 'STOP'

POSITION.

4.1.4 PRESS 'START'

THE PROGRAM WILL IDENTIFY ITSELF AND WILL BEGIN A DIALOGUE WITH THE OPERATOR TO DETERMINE DRIVES TO BE TESTED (SEE SEC 4.5).

THE PROGRAM BEGINS TESTING ONLY THOSE DRIVES SPECIFIED BY THE INPUT DIALOGUE. IF A SPECIFIED DRIVE CANNOT BE FOUND BY THE PROGRAM IT WILL BE FLAGGED AS AN ERROR THAT THE DRIVE WAS NOT AVAILABLE. THEN BEGINNING WITH THE LOWEST NUMERICAL DRIVE AND PROCEEDING IN SEQUENTIAL ORDER, ALL VALID DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE WILL BE

PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS. "END OF PASS" WILL BE TYPED AFTER TESTING ALL DRIVES.

4.2 STARTING LOCATIONS

LOCATION 200 - STARTING ADDRESS TO DEFAULT THE BUSS ADDRESS & THE CONTROLLER INTERRUPT VECTOR & TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. BOTH PORTS SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

LOCATION 220 - STARTING ADDRESS TO INPUT TESTING PARAMETERS VIA THE INPUT DIALOGUE. BUSS ADDRESS & CONT. INTERRUPT VECTOR INPUTTED ONLY ON 1ST PASS.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

THE PROGRAM WILL DETERMINE WHETHER THE DRIVE IS AN RK06/RK07 WITHOUT OPERATOR INPUTS.

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH FUNCTION

```
                                    ------      --------

                                      15     HALT ON ERROR
                                      14     LOOP ON TEST
                                      13     INHIBIT ERROR TYPEOUT
                                      12     BYPASS DRIVE AFTER 20 ERRORS

                                      11     INHIBIT ITERATION
                                      10     BELL ON ERROR
                                       9     LOOP ON ERROR
                                       8     LOOP ON TEST IN SW<07:00>


                              4.3.1   SW<15>

                                      THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT
                                      THE ERROR MESSAGE AND PERTINENT INFORMATION.
                                      PRESSING "CONTINUE" CONTINUES OPERATION OF THE PROGRAM.


                              4.3.2   SW<14>

                                      THE PROGRAM LOOPS ON THE TEST THAT IS BEING EXECUTED WHEN
                                      THE SWITCH IS PUT ON.  THIS SWITCH IS NORMALLY USED ALONG
                                      WITH SW15.


                              4.3.3   SW<13>

                                      THIS SWITCH INHIBITS ALL ERROR MESSAGES.  NORMALLY USED WHEN
                                      LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).
                                      WITH SW<13>SET, SW<15> SHOULD NOT BE SET


                              4.3.4   SW<12>

                                      THIS SWITCH BYPASSES A GIVEN DRIVE AFTER 20 ERRORS HAVE
                                      BEEN DETECTED.


                              4.3.5   SW<11>

                                      EACH TEST WILL BE EXECUTED ONLY ONCE.  NORMALLY AFTER THE
                                      FIRST PASS, EACH SUBTEST IS ITERATED A NUMBER OF TIMES
                                      (USUALLY 50, 5 IN SOME CASES).  SETTING THIS SWITCH INHIBITS
                                      ITERATIONS, SO THAT QUICK PASSES CAN BE MADE.


                              4.3.6   SW<10>

                                      RINGS A BELL ON ERROR.  USEFUL WHEN ERROR TYPEOUT IS INHIBITED.


                              4.3.7   SW<09>

                                      THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR
```

ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK
TO THE BEGINNING OF TEST.

4.3.8 SW<08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST (AS PER
SW<00-7>) FOR EXECUTION AND SUBSEQUENT LOOPING. THUS IF
TEST 15 IS TO BE SELECTED THE SWITCH SETTING WOULD BE
000415. IT SHOULD BE NOTED THAT BEFORE SELECTING & LOOPING
TEST 15, ALL THE PREVIOUS TESTS (1-14) WILL BE EXECUTED.

4.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR
(I.E. AN 11/04 OR 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' AT ANY TIME
EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING
AN RK06 INTERRUPT. 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., 'RUBOUT' 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.

4.5 INPUT DIALOGUE

THE DIALOGUE WILL BE DONE INTERACTIVELY. THE PROGRAM WILL
REQUEST A PARAMETER BY CONSOLE TYPEOUT. THE PARAMETER MAY
THEN BE ENTERED AS SPECIFIED BELOW OR ALLOWED TO DEFAULT
BY A CARRIAGE RETURN. UNRECOGNIZED OR ILLEGAL RESPONSES WILL
BE ECHOED BACK FOLLOWED BY "?". THE PROPER RESPONSE MAY
THEN BE ENTERED.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT

CONSIDERATIONS IN SECTIONS 3.2 & 3.4.

```
538
539                                4.5.1   DRIVE SELECTION
540
541                                        THE REQUEST WILL BE:
542
543                                                DRIVES TO BE TESTED:
544
545                                        THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES
546                                        IN THE 'DRIVE PRESENT' CONDITION.
547
548                                        THE OPERATOR CAN ALSO TYPE IN THE SPECIFIC DRIVE NUMBERS
549                                        TO BE TESTED, SEPARATED BY COMMAS & TERMINATED BY A CARRIAGE
550                                        RETURN.
551
552                                                E.G.  DRIVES TO BE TESTED (EVEN NOS. ONLY):  0,4
553
554                                        NOTES:  1.  FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
555                                                    CONSIDERATIONS IN SECTIONS 3.2 & 3.3.
556
557                                                2.  SEE 'RESTRICTIONS & OPERATOR ACTION' IN SECTION 2.3.
558
559
560                                4.5.2   BUS ADDRESS
561
562                                        THE REQUEST WILL BE:
563
564                                                TYPE IN BUSS ADDRESS IF NOT 177440
565
566                                        THE DEFAULT IS A CARRIAGE RETURN
567
568
569                                4.5.3   CONTROLLER INTERRUPT VECTOR
570
571                                        THE REQUEST WILL BE:
572
573                                                TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210
574
575                                        THE DEFAULT IS A CARRIAGE RETURN.
576
577
578                                4.5.4   EXAMPLE OF PROGRAM DIALOGUE
579
580                                        THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220.
581                                        ALL OPERATOR RESPONSES ARE UNDERLINED.
582
583                                                UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC
584
585                                                MAINDEC-11-DZR6G-A-PB
586
587                                                DRIVES TO BE TESTED:  0,4<CR>
588                                                                      -------
589
590                                                TYPE IN BUSS ADDRESS IF NOT 177440 <CR>
591                                                                                   ----
592
593
```

```
                        TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>
                                                                    ----


                        WILL TEST DRIVES:
                        0
                        4

                        DRIVE 0

                        (THE REST IS IDENTICAL TO THE EXAMPLE SHOWN IN 4.6 BELOW)


4.6     PROGRAM EXAMPLE

THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE
DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.

        UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC

        MAINDEC-11-DZR6G-A-PB

        WILL TEST DRIVES:
        0
        4

        DRIVE 0

        DRIVE SERIAL NO. AAA
        CARTRIDGE SERIAL NO  BBB

        DRIVE 4

        DRIVE SERIAL NO. CCC
        CARTRIDGE SERIAL NO. DDD

        END PASS #1


        WILL TEST DRIVES:
        0
        4

        DRIVE 0


        DRIVE 4

        END PASS # 2

           (ETC)

THE ABOVE ASSUMES NO ERRORS DETECTED.
THE NUMBER OF PASSES IS DETERMINED BY ACT/APT/XXDP

NOTES:  1.  FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
            CONSIDERATIONS IN SECTIONS 3.2 & 3.3.
```

```
650
651                                     2.  SEE 'RESTRICTIONS & OPERATOR ACTION' IN SECTION 2.3.
652
653                             4.7     HALTING THE PROGRAM
654
655                                     THE PROGRAM PROVIDES A METHOD OF HALTING ITSELF SUCH THAT
656                                     THE CARTRIDGE AND/OR DRIVE IS NOT LEFT IN ON UNDETERMINED
657                                     STATE; IE: HEADS UNLOADED OR INVALID FORMAT.
658
659                                     TO PROPERLY HALT, TYPE CONTROL-C ( C) ON THE CONSOLE.
660
661                                     IF HEADS ARE LOADED, THE PROGRAM WILL:
662
663                                     1.  ECHO  C
664                                     2.  TYPE "CPU HALTED"
665                                     3.  HALT THE PROGRAM
666
667                                     IF HEADS ARE NOT LOADED, THE PROGRAM WILL:
668
669                                     1.  ECHO  C
670                                     2.  TYPE 'HALT PENDING, PLEASE WAIT'
671                                     3.  WILL LOAD HEADS
672                                     4.  TYPE 'CPU HALTED'
673                                     5.  HALT THE PROGRAM
674
675
676                                     NOTES:
677
678                                     1.  OPERATING THE 'CONTINUE' SWITCH ON THE CPU CONSOLE WILL RETURN THE
679                                         PROGRAM TO TEST 1 WHERE TESTING WILL BEGIN WITH THE 1'ST DRIVE AGAIN.
680
681
682
683                             5.0     DUAL PORT DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION
684
685                             5.1     GENERAL
686
687                                     A.  BASIC CONTROLLER TESTS, SIZING & SETUP
688
689                                         THESE TESTS DO BASIC CONTPOLLER REGISTER REFERENCE TESTS,
690                                         CHECKS OPERATOR INPUTS AGAINST DRIVES ON THE LINE OR DE-
691                                         FAULTS TO TEST ALL DRIVES SEEN ON THE LINE WITHIN THE
692                                         RESTRICTIONS DESCRIBED IN SECTION 2.3.  CHECKS ARE MADE ON
693                                         THE EXISTENCE OF EITHER AN L OR P CLOCK.
694
695                                     B.  DUAL PORT TESTS
696
697                                         THESE TESTS VERIFY THE ABILITY OF THE DRIVE TO OPERATE
698                                         IN THE DUAL PORT MODE.  RELEASE, TIMEOUT & PROPER INTER-
699                                         ACTION BETWEEN THE PORTS ARE VERIFIED.
700
701                                     C.  METHOD TO DETERMINE THAT THE DRIVE IS IN NEUTRAL OR SEIZED.
702
703                                             THE PROGRAM DOES A 'SELECT DRIVE' COMMAND TO PORT 'A'.  IF
704                                             MESSAGE A0 RETURNS WITH THE 'DRIVE AVAILABLE' BIT NOT SET,
705                                             IT ASSUMES PORT 'B' HAS SEIZED THE DRIVE.
```

IF MESSAGE A0 RETURNS WITH THE 'DRIVE AVAILABLE' BIT SET,
IT ASSUMES THE DRIVE WAS IN NEUTRAL OR ALREADY SEIZED BY
PORT 'A'.

AFTER A TIMEOUT OF 1 SECOND BY PORT 'A', ASSUMING NO FURTHER
COMMANDS, THE DRIVE SHOULD BECOME AVAILABLE TO PORT 'B'.

THERE IS NO SPECIFIC METHOD TO VERIFY THE DRIVE IS IN NEUTRAL
BECAUSE THE ACT OF DOING A SELECT DRIVE COMMAND TO CHECK FOR
NEUTRAL SEIZES THE DRIVE & TAKES IT OUT OF NEUTRAL.

'DSC' & 'ATTN' DO NOT ASSERT AT THE END OF A TIMEOUT IF THERE
IS NO PREVIOUS PORT REQUEST MADE.

THEY WILL ASSERT, HOWEVER, IF A PORT REQUESTED WHILE SEIZED
BY THE OTHER PORT & A TIMEOUT OR RELEASE OCCURS ON THE OTHER
PORT.

5.2 TEST DESCRIPTIONS

****
BASIC CONTROLLER TESTS, SIZING & SETUP
****

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL
RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY
ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
TESTS AND JUMPING TO 'END OF PASS'

TEST 2 SIZE THE BUSS

THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
MANUAL MODE.
EVERY EVEN NUMBERED DRIVE (0,2,4,6) IS ADDRESSED.
CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
DRIVE WILL BE TESTED AS AN RK06. IF SET, THE PROGRAM WILL BYPASS
TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
DICATING THE OTHER PORT IS ACCESSED.
IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS RK07.

TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
DEFAULTED. EVERY EVEN NUMBERED DRIVE IS ADDRESSED &
CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06
IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH

```
                        NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                        NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFORMATION TO
                        VERIFY IT WAS NOT SPECIFIED.
                        IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS RK07.


                    TEST 4  FIND NEXT DRIVE TO BE TESTED

                        THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                        ADDRESS IN '$UNIT' & $TMP4 IS SET TO CDT IF DRIVE IS RK07.
                        THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                        THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.



                    ;*******************************************************************
                    DUAL PORT TESTS
                    ;*******************************************************************

                    TEST 5  TEST PORT 'A' SEIZE & TIMEOUT

                        VERIFY THAT THE DRIVE CAN BE SEIZED & THAT THE PORT
                        TIMEOUT RELEASES THE DRIVE.

                        A.  SET VOLUME VALID FOR BOTH PORTS & DO A RECAL COMMAND

                        B.  A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'A'.
                            THE PROGRAM VERIFIES THE DRIVE HAS BEEN SEIZED BY 'DRIVE
                            AVAILABLE' SET.

                        C.  A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'B' THE
                            PROGRAM VERIFIES THAT 'DRIVE AVAILABLE' IS NOT SET
                            FOR PORT 'B' & THAT CERR IS SET.

                        D.  VERIFY THAT FOR ALL MESSAGES REQUESTED THRU PORT 'B', (MSG
                            A0-A3, B0-B3) THAT MESSAGE 0 ALWAYS RETURNS FROM PORT B
                            WHILE PORT 'A' IS SEIZED.

                        E.  WAIT FOR THE PORT TIMEOUT TO OCCUR ON PORT 'A' BY
                            CONTINUOUSLY CHECKING FOR ATTN ON PORT B.  AFTER
                            ATTN-B IS REC'D, A DRIVE SELECT COMMAND IS ISSUED
                            THRU PORT B & "DRIVE AVAILABLE" IS CHECKED TO
                            BE SET IN MESSAGE A0.

                            MEASURE THE DURATION OF THE TIMEOUT & TYPE THE VALUE
                            FOR THE FIRST PASS ONLY.

                        F.  VERIFY THAT ONLY PORT 'B' GETS 'DSC' & 'ATTN'.

                        G.  VERIFY THE DRIVE CLEAR COMMAND CLEARS 'DSC' & 'ATTN'
                            ON PORT 'B' BUT DOES NOT RELEASE THE DRIVE FROM PORT 'B'.


                    TEST 6  TEST PORT 'B' SEIZE & TIMEOUT

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'
```

```
                TEST 7  PRINT DRIVE SERIAL NUMBER

                        THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A3
                        IN BCD ON THE 1'ST PASS ONLY.
                        IT ALSO TESTS THAT THE SERIAL # READ THRU BOTH PORTS
                        ARE THE SAME.


                TEST 10 TEST PORT 'A' COMMAND SEIZE & ATTENTION

                        VERIFY THE OPERATION OF 'DSC' & 'ATTN' BITS AFTER A COMMAND.

                        A.  ISSUE A SEEK COMMAND TO CYLINDER 10 THRU PORT 'A'.


                        B.  VERIFY SEIZURE & THAT 'DSC' & 'ATTN' SETS FOR PORT 'A'
                            ONLY AFTER THE SEEK HAS COMPLETED.

                        C.  VERIFY 'ATTN' REMAINS SET BEYOND TIMEOUT

                        D.  VERIFY A DRIVE CLEAR COMMAND RESETS 'DSC' & 'ATTN'
                            & DOES NOT RELEASE THE DRIVE FROM PORT 'A'.


                TEST 11 TEST PORT 'B' COMMAND SEIZE & ATTENTION

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'
                        BUT THE SEEK IS TO CYLINDER 0.


                TEST 12 TEST RESET PORT 'A' ATTENTION BY DRIVE CLEAR COMMAND

                        VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF
                        THE SEIZING PORT

                        A.  SET EACH PORT'S ATTENTION BIT BY PERFORMING SEEK
                            COMMANDS TO CYLINDER 0 & ALLOWING TIMEOUTS.

                        B.  SEIZE THE DRIVE THRU PORT 'A' & ISSUE A DRIVE CLEAR COMMAND
                            VERIFY THAT 'DSC' & 'ATTN' FOR PORT 'A' HAVE BEEN CLEARED

                        C.  SEIZE THE DRIVE THRU PORT 'B' & VERIFY 'DSC' & 'ATTN'
                            HAVE NOT CLEARED

                TEST 13 TEST RESET PORT 'B' ATTENTION BY DRIVE CLEAR COMMAND

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


                TEST 14 TEST RELEASE, DRIVE SEIZED BY PORT 'A'

                        A.  SEIZE THE DRIVE THRU PORT 'A'
```

```
                        B.  ISSUE A RELEASE USING RKCS2 THRU PORT 'A'

                        C   VERIFY PORT 'B' CAN ACCESS THE DRIVE IMMEDIATELY &
                            THAT NEITHER PORT SEES 'DSC' OR 'ATTN'


TEST 15 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 16 TEST RELEASE FROM PORT 'A' WITH PORT 'B' REQUESTING

                        A.  PORT 'A' SEIZES THE DRIVE & DOES A SEEK TO SELF COMMAND.
                            THE PROGRAM VERIFIES 'DSC' & 'ATTN' ON PORT 'A' ONLY
                            ON COMPLETION

                        B.  PORT 'B' TRIES TO ACCESS THE DRIVE.  THE PROGRAM VERIFIES
                            DRIVE NOT AVAILABLE

                        C.  A RELEASE BY PORT 'A' IS ISSUED.  VERIFY PORT 'B' CAN
                            ACCESS THE DRIVE IMMEDIATELY & THAT 'DSC' & 'ATTN'
                            ARE SEEN ON PORT 'B' .

                        D.  VERIFY PORT 'A' 'DSC' & 'ATTN' REMAINS SET AFTER RELEASE

                        E.  THE PROGRAM ISSUES A DRIVE CLEAR COMMAND TO PORT 'B'
                            & VERIFIES 'DSC' & 'ATTN' RESETS.

                        F.  THE PROGRAM THEN VERIFIES THAT PORT 'B' DOES NOT SEE
                            FURTHER (MULTIPLE) ATTENTIONS FROM WHAT WOULD HAVE BEEN
                            NORMAL TIMEOUT FROM PORT 'A'.


TEST 17 TEST RELEASE FROM PORT 'B' WITH PORT 'A' REQUESTING

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 20 TEST RELEASE FROM REQUESTING PORT 'B' INHIBITS 'ATTN'

                        A.  PORT 'A' SEIZES THE DRIVE
                        B.  PORT 'B' ATTEMPTS TO SEIZE THE DRIVE
                        C.  PORT 'B' & PORT 'A' RELEASE THE DRIVE, IN THAT ORDER
                        D.  THE PROGRAM VERIFIES THAT NEITHER PORT 'A' OR 'B' ATTENTION
                            BITS SET


TEST 21 TEST RELEASE FROM REQUESTING PORT 'A' INHIBITS 'ATTN'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 22 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
```

```
                        VERIFY THAT A RELEASE ISSUED BY ONE PORT IS NOT RECOGNIZED IF
                        THE DRIVE IS SEIZED BY THE OTHER PORT

                        A.  SEIZE THE DRIVE THRU PORT 'A'.

                        B.  ISSUE A RELEASE THRU PORT 'B' & VERIFY DRIVE STILL SEIZED
                            BY PORT 'A'.


TEST 23 TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 24 TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE FROM PORT 'A'

                        A.  ISSUE A SEEK COMMAND TO CYL 10 FROM PORT 'A' &
                            AN IMMEDIATE RELEASE TO PORT A.

                        B.  VERIFY THE DRIVE IS AVAILABLE TO PORT 'B'
                            & PORT 'B' RECEIVES 'ATTN'

                        C.  VERIFY PORT 'A' DOES NOT RAISE ATTN WHEN THE SEEK IS
                           COMPLETED.


TEST 25 COMMAND & IMMEDIATE RELEASE FROM PORT 'B'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'
                        BUT THE SEEK IS TO CYLINDER 0.


TEST 26 TEST TIMEOUT RETRIGGER THRU PORT 'A'

                        VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.

                        A.  PORT 'A' SEIZES THE DRIVE

                        B.  THE PROGRAM WAITS 500MS & RE-SEIZES THE DRIVE THRU PORT 'A'

                        C.  PORT 'B' ATTEMPTS TO SEIZE THE DRIVE & THE
                            PROGRAM VERIFIES THAT FULL TIMEOUT TOOK PLACE
                            FROM STEP 'B' ABOVE.


TEST 27 TEST TIMEOUT RETRIGGER THRU PORT 'B'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 30 TEST PORT 'A' TIMER INHIBIT

                        A.  PORT 'A' SEIZES THE DRIVE
                        B.  PORT 'B' ATTEMPTS TO SEIZE THE DRIVE
```

```
                        C.  PORT 'A' RELEASES THE DRIVE
                        D.  PORT 'A' ATTEMPTS TO GET THE DRIVE BACK.

                        THE PROGRAM VERIFIES THAT PORT 'A' CANNOT ACCESS
                        THE DRIVE FOR APPROX 1 SEC



TEST 31 TEST PORT 'B' TIMER INHIBIT

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 32 TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'A'

                        VERIFY THAT THE UNLOAD COMMAND THRU A PORT, SEIZES THAT
                        PORT FOR AS LONG HAS HEADS ARE UNLOADED & RELEASE IS NOT
                        ISSUED.

                        A.  ISSUE AN UNLOAD COMMAND THRU PORT 'A'.
                            VERIFY DRIVE UNLOADS & ATTENTION IS SET.

                        B.  DELAY FOR 5 SECONDS & VERIFY DRIVE
                            NOT AVAILABLE TO PORT 'B' TO INSURE TIMERS INHIBITED.

                        C.  ISSUE A RELEASE FROM PORT 'A'.  VERIFY DRIVE BECOMES
                            AVAILABLE TO PORT 'B'

                        D.  LOAD HEADS FROM PORT 'B' & VERIFY 'ATTN-B' AT COMPLETION


TEST 33 TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'B'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 34 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'A'

                        VERIFY THAT THE RECAL COMMAND THRU A PORT, SEIZES THAT
                        PORT FOR AS LONG AS THE RECAL IS IN PROGRESS & A
                        RELEASE IS NOT ISSUED.

                        A.  ISSUE A RECAL COMMAND FROM THE LAST CYL THRU PORT 'A'

                        B.  VERIFY PORT 'B' CANNOT SEIZE THE DRIVE UNTIL
                            PORT 'A' RECEIVES ATTN.
                            THIS INSURES THAT THE TIMERS WERE INHIBITED


TEST 35 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'B'

                        THE PREVIOUS TEST IS REPEATED FOR PORT 'B'


TEST 36 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
```

```
1042
1043                    THIS TEST VERIF!ES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
```

```
                                                    THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
                                                    FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
                                                    AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.

                                                    SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED
                                                    SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE

                                                    IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
                                                    IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,

                                                    A MESSAGE WILL BE TYPED INDICATING THAT ALL
                                                    FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
                                                    THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITI

                                                    THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.


                                                TEST 37 DATA TESTS

                                                    VERIFY UNIQUE DATA CAN BE WRITTEN THRU EITHER PORT & READ
                                                    BACK CORRECTLY THRU BOTH PORTS.

                                                    A.  ALL 1'S ARE WRITTEN THRU PORT 'A' ON CYL 0, SECTOR 0,
                                                        TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.

                                                    B.  ALL 1'S ARE WRITTEN THRU PORT 'B' ON CYL 10, SECTOR 0,
                                                        TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.

                                                    C.  THE PROGRAM CHECKS THAT CYL 0 WAS NOT OVERWRITTEN
                                                        BY READING  & VERIFING ALL 0'S THRU PORT 'B'.

                                                TEST 40 ALTERNATING SEEK INTERACTION TEST

                                                    THIS TEST VERIFIES THAT THERE ARE NO TIMING INTERACTION PROBLEMS
                                                    BETWEEN SEEKS FROM BOTH PORTS.

                                                    A.  PORT 'A' SEIZES THE DRIVE & SEEKS TO CYLINDER 0 & RELEASES
                                                        THE DRIVE AFTER 'ATTN' IS RECEIVED.
                                                        THE PROGRAM VERIFIES THAT UNTIL ATTN IS REC'D,
                                                        PORT 'B' SEES 'CONTROLLER ERROR' & 'DRIVE NOT AVAILABLE'.

                                                    B.  PORT 'B' SEIZES THE DRIVE & SEEKS TO THE LAST CYL
                                                        & RELEASES THE DRIVE AFTER 'ATTN' IS RECEIVED
                                                        THE PROGRAM VERIFIES THAT UNTIL ATTN IS REC'D,
                                                        PORT 'A' SEES 'CONTROLLER ERROR' & 'DRIVE NOT AVAILABLE'.

                                                    C.  THE ABOVE IS REPEATED FOR A PATTERN OF CONVERGING SEEKS
                                                        TOWARD THE CENTER OF THE CARTRIDGE.

                                                    D.  THE PROGRAM VERIFIES MULTIPLE ATTENTIONS OR ERRORS
                                                        DO NOT OCCUR AS A RESULT OF TIMING PROBLEMS.


                                    6.0     ERROR REPORTING
```

6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN. MSG A(00), MSG B(01), RKER, RKBA...ETC, INDICATE THE CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF ERROR.

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVISED TO LOOK UP THE PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY

VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR MESSAGES ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLE:

```
        MESSAGE A0 ERROR
        AFTER START SPINDLE CMD & FWD SET

          TEST NO.   PC
          000014  016530
        EXPECT
A0      B0      A1      B1      A2      B2      B3
030144  100000  013704  000001
        ACTUAL
  140144  100000  101744  000001
RKCS1   RKCS2   RKASOF  RKER    RKDS    RKDC
040200  000100  010000  000000  000000  000000
```

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR MESSAGE REGISTERS A0, B0, A1 & B1.

MESSAGES A2, B2 & B3 WILL BE TYPED OUT ONLY AS REQUIRED IF THE CYLINDER DIFFERENCE/OFFSET, CYLINDER ADDRESS & HEAD & SECTOR INFORMATION IS A VARIABLE PARAMETER OF THE TEST.

7.0 DUAL PROCESSOR-DUAL CONTROLLER TESTING

```
1156                    THIS PROGRAM AS DESCRIBED, TESTS THRU A SINGLE CPU & RK611
1157                    IN A SYNCHRONOUS MANNER.
1158
1159                    TO RUN IN A DYNAMIC MANNER, I.E.: DUAL CPU/DUAL RK611
1160                    WITH BOTH PORTS CONTENDING FOR THE DRIVE ASYNCHRONOUSLY,
1161                    THE FOLLOWING PROGRAMS SHOULD BE USED:
1162
1163                    DZR6M   SUBSYSTEM VERIFICATION PART 1  TEST 22
1164                    DZR6P   PERFORMANCE EXERSICER
1165
1166                    EACH OF THE ABOVE PROGRAMS REQUIRES A SPECIAL STARTING ADDRESS
1167                    TO ENTER THE DUAL PORT TESTING MODE.
1168
1169                    TEST 22 IN THE SUBSYSTEM VERIFICATION PROGRAM CAN BE SET UP
1170                    SO THAT ONE PORT BECOMES THE WRITER OF A SPECIFIED DATA PATTERN
1171                    OVER A SPECIFIED RANGE OF HEADS & CYLINDERS, WHILE THE
1172                    OTHER PORT BECOMES THE READER.
1173                    THE ROLES OF EACH PORT CAN THEN BE REVERSED THRU ANOTHER SETUP.
1174
1175                    THERE ARE MANY POSSIBLE WAYS OF DYNAMICALLY TESTING DUAL
1176                    PORT BY THE ABOVE TEST.
1177                    THE OPERATOR IS URGED TO READ THE PROGRAM DOCUMENT.
1178
1179                    THE PERFORMANCE EXERCISER SHOULD BE USED TO RANDOMLY
1180                    TEST ALL POSSIBLE DUAL PORT OPERATIONS.
1181                    THIS PROGRAM WILL MOST SIMULATE ACTUAL ON-LINE DUAL PORT OPERATION.
1182                    THE OPERATOR IS URGED TO READ THE PROGRAM DOCUMENT.
1183                                    END OF SPECIFICATION
1184
```

```
                        ; *** PGM REV 030 ***
                        ;  29-SEP-77 PATCH ALL ERROR LOGS

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

                        ;DEFINE SYSMAC MACROS

167400                  $SWR=   167400                  ;DEFINE SWITCHES 15,14,13,11,10,9,8
000001                  $TN=    1                       ;SET FIRST TEST NO.  TO 1


                        .TITLE  UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC
                        ;*COPYRIGHT (C) 1976,1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY GARY PAPAZIAN
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*

                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            ABORT DRIVE AFTER 20 ERRORS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>


                        .SBTTL  SUMMARY OF STARTING LOCATIONS
                        ;*
                        ;*
                        ;*              200     DEFAULT PARAMETERS
                        ;*              220     INPUT PARAMETERS
                        ;*              240     ODT11
                        ;*
```

```
                        .SBTTL  BASIC DEFINITIONS

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

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

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

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

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

```
                    000002          SW01=   2
                    000001          SW00=   1
                                    .EQUIV  SW09,SW9
                                    .EQUIV  SW08,SW8
                                    .EQUIV  SW07,SW7
                                    .EQUIV  SW06,SW6
                                    .EQUIV  SW05,SW5
                                    .EQUIV  SW04,SW4
                                    .EQUIV  SW03,SW3
                                    .EQUIV  SW02,SW2
                                    .EQUIV  SW01,SW1
                                    .EQUIV  SW00,SW0

                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                    100000          BIT15=  100000
                    040000          BIT14=  40000
                    020000          BIT13=  20000
                    010000          BIT12=  10000
                    004000          BIT11=  4000
                    002000          BIT10=  2000
                    001000          BIT09=  1000
                    000400          BIT08=  400
                    000200          BIT07=  200
                    000100          BIT06=  100
                    000040          BIT05=  40
                    000020          BIT04=  20
                    000010          BIT03=  10
                    000004          BIT02=  4
                    000002          BIT01=  2
                    000001          BIT00=  1
                                    .EQUIV  BIT09,BIT9
                                    .EQUIV  BIT08,BIT8
                                    .EQUIV  BIT07,BIT7
                                    .EQUIV  BIT06,BIT6
                                    .EQUIV  BIT05,BIT5
                                    .EQUIV  BIT04,BIT4
                                    .EQUIV  BIT03,BIT3
                                    .EQUIV  BIT02,BIT2
                                    .EQUIV  BIT01,BIT1
                                    .EQUIV  BIT00,BIT0

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

                                     SBTTL  RK06 CONTROLLER REGISTER DEFINITION
```

```
1343
1344                            ;       $BASE=177440
1345
1346    000000                  RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1347    000002                  RKWC=   2               ;WORD COUNT REGISTER
1348    000004                  RKBA=   4               ;BUS ADDRESS REGISTER
1349    000006                  RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1350    000010                  RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1351    000012                  RKDS=   12              ;DRIVE STATUS REGISTER
1352    000014                  RKER=   14              ;ERROR REGISTER
1353    000016                  RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1354    000020                  RKDC=   20              ;DESIRED CYLINDER REGISTER
1355    000024                  RKDB=   24              ;DATA BUFFER
1356    000026                  RKMR1=  26              ;MAINTENANCE REGISTER 1
1357    000034                  RKMR2=  34              ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1358    000036                  RKMR3=  36              ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1359    000030                  RKECPS= 30              ;ECC POSITION INFORMATION
1360    000032                  RKECPT= 32              ;ECC PATTERN INFORMATION
1361
1362                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1363
1364                            ;       DRIVE COMMANDS
1365
1366    000001                  SELDRV= 1               ;SELECT DRIVE (GET STATUS)
1367    000003                  PACK=   3               ;PACK ACKNOWLEDGE
1368    000005                  CLEAR=  5               ;DRIVE CLEAR
1369    000007                  UNLOAD= 7               ;UNLOAD
1370    000011                  SRTSPL= 11              ;START SPINDLE
1371    000013                  RECAL=  13              ;RECALIBRATE
1372    000015                  OFFSET= 15              ;OFFSET
1373    000017                  SEEK=   17              ;SEEK
1374    000021                  RDDATA= 21              ;READ DATA
1375    000023                  WRDATA= 23              ;WRITE DATA
1376    000025                  RDHEAD= 25              ;READ HEADER
1377    000027                  WRHEAD= 27              ;WRITE HEADER AND DATA
1378    000031                  WRTCHK= 31              ;WRITE CHECK
1379
1380    000001                  GO=     BIT0            ;GO BIT
1381    000100                  IE=     BIT6            ;INTERRUPT ENABLE
1382    000200                  RDY=    BIT7            ;CONTROLLER READY
1383    000400                  BA16=   BIT8            ;BUS ADDRESS BIT 16
1384    001000                  BA17=   BIT9            ;BUS ADDRESS BIT 17
1385    002000                  CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
1386    004000                  CTO=    BIT11           ;CONTROLLER TIMEOUT
1387    010000                  CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1388    020000                  DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1389    040000                  DI=     BIT14           ;DRIVE INTERRUPT
1390    100000                  CERR=   BIT15           ;CONTROLLER ERROR
1391    100000                  CCLR=   BIT15           ;CONTROLLER CLEAR
1392
1393                            .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1394
1395    000007                  DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
1396    000010                  RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1397    000020                  BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
1398    000040                  SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
1399            000100                  IR=     BIT6                    ;INPUT READY
1400            000200                  OR=     BIT7                    ;OUTPUT READY
1401            000400                  UFE=    BIT8                    ;UNIT FIELD ERROR
1402            001000                  MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1403            002000                  PGE=    BIT10                   ;PROGRAMMING ERROR
1404            004000                  NEM=    BIT11                   ;NON-EXISTENT MEMORY
1405            010000                  NED=    BIT12                   ;NON-EXISTENT DRIVE
1406            020000                  UPE=    BIT13                   ;UNIBUS PARITY ERROR
1407            040000                  WCE=    BIT14                   ;WRITE CHECK ERROR
1408            100000                  DLT=    BIT15                   ;DATA LATE ERROR
1409
1410                                    .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
1411
1412            000001                  ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1413            000002                  SKI=    BIT1                    ;SEEK INCOMPLETE
1414            000004                  NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
1415            000010                  DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
1416            000020                  FMTE=   BIT4                    ;FORMAT ERROR
1417            000040                  DTYE=   BIT5                    ;DRIVE TYPE ERROR
1418            000100                  ECH=    BIT6                    ;ECC HARD
1419            000200                  BSE=    BIT7                    ;BAD SECTOR ERROR
1420            000400                  HVRC=   BIT8                    ;HEADER VRC ERROR
1421            001000                  COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1422            002000                  IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR:  HEAD/CYL
1423            004000                  WLE=    BIT11                   ;WRITE LOCK ERROR
1424            010000                  DTE=    BIT12                   ;DRIVE TIMING ERROR
1425            020000                  OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1426            040000                  UNS=    BIT14                   ;DRIVE UNSAFE
1427            100000                  DCK=    BIT15                   ;DATA CHECK
1428
1429                                    .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
1430
1431            000001                  DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1432                                                                    ;    THIS BIT IS RESET)
1433            000004                  OFST=   BIT2                    ;DRIVE OFFSET
1434            000010                  ACLO=   BIT3                    ;AC LOW
1435            000020                  DCLO=   BIT4                    ;DC LOW
1436            000040                  DROT=   BIT5                    ;DRIVE OFF TRACK
1437            000100                  VV=     BIT6                    ;VOLUME VALID
1438            000200                  DRDY=   BIT7                    ;DRIVE READY
1439            000400                  DDT=    BIT8                    ;DRIVE TYPE (0=RK06,1=RK07)
1440            004000                  WRL=    BIT11                   ;WRITE LOCK
1441            020000                  PIP=    BIT13                   ;POSITIONING IN PROGRESS
1442            040000                  DSC=    BIT14                   ;DRIVE STATUS CHANGE
1443            100000                  SVAL=   BIT15                   ;STATUS VALID
1444
1445                                    .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
1446
1447            000017                  MESMSK= 17                      ;MESSAGE MASK
1448            000020                  PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
1449            000040                  DMD=    BIT5                    ;DIAGNOSTIC MODE
1450            000100                  MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1451            000200                  MIND=   BIT7                    ;MAINTENANCE INDEX
1452            000400                  MCLK=   BIT8                    ;MAINTENANCE CLOCK
1453            001000                  MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1454            002000                  MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
```

```
1455            004000                  PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
1456            010000                  PCD=    BIT12                   ;PRECOMPENSATION DELAY
1457            020000                  ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
1458            040000                  WRTGAT= BIT14                   ;WRITE GATE
1459            100000                  RDGATE= BIT15                   ;READ GATE
1460
1461                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1462
1463            000040                  D.DRA=  BIT5                    ;DRIVE AVAILABLE
1464            000100                  D.VV=   BIT6                    ;VOLUME VALID
1465            000200                  D.DRDY= BIT7                    ;DRIVE READY
1466            000400                  D.DDT=  BIT8                    ;DRIVE TYPE (0=RK06,1=RK07)
1467            001000                  D.FORM= BIT9                    ;DRIVE FORMAT
1468            002000                  D.OFF=  BIT10                   ;OFFSET ON
1469            004000                  D.WRL=  BIT11                   ;WRITE LOCK
1470            010000                  D.SPIN= BIT12                   ;SPINDLE ON
1471            020000                  D.PIP=  BIT13                   ;POSITIONING IN PROGRESS
1472            040000                  D.DSC=  BIT14                   ;DRIVE STATUS CHANGE
1473
1474                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1475
1476            000020                  D.SSP=  BIT4                    ;SERVO SIG PRESENT
1477            000040                  D.HDHM= BIT5                    ;HEADS HOME
1478            000100                  D.BRHM= BIT6                    ;BRUSHES HOME
1479            000200                  D.DOOR= BIT7                    ;DOOR INTERLOCKED
1480            000400                  D.CART= BIT8                    ;CARTRAGE INTERLOCK
1481            001000                  D.SPOK= BIT9                    ;SPEED OK
1482            002000                  D.FWD=  BIT10                   ;FORWARD
1483            004000                  D.REV=  BIT11                   ;REVERSE
1484            010000                  D.LOAD= BIT12                   ;HEADS LOADING
1485            020000                  D.RTZ=  BIT13                   ;RETURN TO ZERO
1486            040000                  D.UNLD= BIT14                   ;HEADS UNLOADING
1487
1488                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1489
1490            000040                  D.IDAE= BIT5                    ;INVALID DISK ADDRESS ERROR: HEAD/CYL
1491            000100                  D.ACLO= BIT6                    ;AC LOW
1492            000200                  D.FLT=  BIT7                    ;DRIVE FAULT
1493            000400                  D.ILF=  BIT8                    ;ILLEGAL FUNCTION CODE
1494            001000                  D.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
1495            002000                  D.SKI=  BIT10                   ;SEEK INCOMPLETE
1496            004000                  D.WLE=  BIT11                   ;WRITE LOCK ERROR
1497            010000                  D.SPLS= BIT12                   ;SPEED LOSS
1498            020000                  D.DROT= BIT13                   ;DRIVE OFF TRACK
1499            040000                  D.UNS=  BIT14                   ;R/W UNSAFE
1500
1501                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1502
1503            000020                  D.SECT= BIT4                    ;SECTOR ERROR
1504            000040                  D.WCUR= BIT5                    ;WRITE CURRENT AND NO WRITE GATE
1505            000100                  D.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
1506            000200                  D.HDFL= BIT7                    ;HEAD FAULT
1507            000400                  D.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
1508            001000                  D.XERR= BIT9                    ;INDEX ERROR
1509            002000                  D.TIB=  BIT10                   ;TRIBIT ERROR
1510            004000                  D.PLO=  BIT11                   ;PLO ERROR
```

```
1511            010000                  D.NMOV= BIT12                   ;SEEK AND NO MOTION
1512            020000                  D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1513            040000                  D.SUNS= BIT14                   ;SERVO UNSAFE
1514
1515                                    .SBTTL  COMMON MASKS AND OTHER BITS:  MESSAGE A (RKMR2:34)
1516
1517            000007                  M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1518            077770                  M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1519
1520                                    .SBTTL  COMMON MASKS AND OTHER BITS:  MESSAGE B (RKMR3:36)
1521
1522            000003                  M.ID=   3                       ;BYTE ID, ALL BYTES
1523            040000                  M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1524            000760                  M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1525            007000                  M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1526            100000                  M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
1527
1528                                    .SBTTL  TRAP CATCHER
1529
1530            000000                          .=0
1531                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1532                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1533                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1534            000174                          .=174
1535  000174    000000                DISPREG:  .WORD  0                ;;SOFTWARE DISPLAY REGISTER
1536  000176    000000                SWREG:    .WORD  0                ;;SOFTWARE SWITCH REGISTER
1537                                    .SBTTL  STARTING ADDRESS(ES)
1538  000200    000137  010050                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1539            000220                          .=220
1540  000220    000137  010040                  JMP     PARSRT          ;INPUT PARAMETERS
1541
1542            000240                          .=240
1543  000240    000137  067302                  JMP     0.ODT           ;ENTER ODT11
1544
1545                                    .SBTTL  ACT11 HOOKS
1546
1547                                    ;;**********************************************************************
1548                                    ;HOOKS REQUIRED BY ACT11
1549            000244                          $SVPC=.                 ;SAVE PC
1550            000046                          .=46
1551  000046    042772                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1552            000052                          .=52
1553  000052    100000                          .WORD   100000          ;;2)SET LOC.52 TO 100000
1554            000244                          .=$SVPC                 ;; RESTORE PC
1555            001000                          .=1000
1556                                    .SBTTL  APT PARAMETER BLOCK
1557
1558                                    ;;**********************************************************************
1559                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1560                                    ;;**********************************************************************
1561            001000                          .$X=.   ;;SAVE CURRENT LOCATION
1562            000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1563  000024    000200                          200     ;;FOR APT START UP
1564            000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1565  000044    001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
1566            001000                          .=.$X   ;;RESET LOCATION COUNTER
1567                                    ;;**********************************************************************
1568                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1569                                    ;INTERFACE SPEC.
1570
1571  001000                            $APTHD:
1572  001000    000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1573  001002    001210                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1574  001004    000430                  $TSTM:  .WORD   280.    ;;RUN TIM OF LONGEST TEST
1575  001006    001130                  $PASTM: .WORD   600.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1576  001010    001130                  $UNITM: .WORD   600.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1577  001012    000042                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1578
1579
1580                                            .LIST   MD
1581
1582                                    :
```

```
1583                            ;USE LOOP X TO OMIT SUBCLR
1584                            ;
1585
1586                            .MACRO  LOOP    A
1587                                    SCOP1
1588                                    MOV     #STACK,SP       ;RESTORE STK PTR
1589
1590                            .IF     B       A
1591                                    JSR     PC,SUBCLR
1592                                    ERROR   24              ;CERR AFTER SCLR
1593
1594                            .ENDC
1595                            .ENDM   LOOP
1596
1597                            ;
1598                            ;THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD BITS
1599                            ;A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1600                            ;NOTE:  A CAN BE ANY BIT COMBINATION DESIRED & INCLUDES D.DRA
1601                            ;
1602                            .MACRO  F.EAB   A
1603
1604                                    MOV     #<A!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
1605                                    CLR     E.B0            ;EXPECTED MSG B0
1606                                    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
1607                                    MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
1608                                    CLR     E.A2            ;EXPECTED MSG A2
1609                                    MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
1610                                    MOV     #3,E.B3         ,MSG ID FOR EXPECTED MSG B3
1611                            .ENDM   F.EAB
1612
1613                            ;
1614                            ;THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
1615                            ;USE    A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
1616                            ;USE    G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1617                            ;       H=T.B2  TO READ MSG B2 & PUT INFO INTO 'CYLADD'
1618                            ;       I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
1619                            ;
1620                            ;USE    F=<ERROR DESCRIPTION>
1621                            ;
1622                            .MACRO  CHECK   A,C,D,E,F,G,H,I
1623
1624                                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
1625                                    .WORD   G!H!I           ;& MSGS SPECIFIED HERE
1626                                    ERROR   A               ;MSG A0 ERROR F
1627                                    ERROR   C               ;MSH B0 ERROR
1628                                    ERROR   D               ;MSG A1 ERROR
1629                                    ERROR   E               ;MSG B1 ERROR
1630                            .ENDM   CHECK
1631
1632                            ;
1633                            ;A=CYL DIFF/OFFSET ERROR #
1634                            ;B=CYL ADDR ERROR #
1635                            ;C=<ERROR DESCRIPTION>
1636                            ;
1637                            .MACRO  CWD2    A,B,C,?D,?E
1638                                    MOV     #2,RKMR1(R5)
```

```
1639                                            JSR     PC,GSTAT
1640                                            TST     CYLDIF          ;SEE IF MSG A2=0
1641                                            BEQ     D               ;BR IF YES
1642                                            ERROR   A               ;MSG A2 NOT CLEARED C
1643                                    D:      TST     CYLADD          ;SEE IF MSG B2=0
1644                                            BEQ     E               ;BR IF YES
1645                                            ERROR   B               ;MSG B2 NOT CLEARED C
1646                                    E:
1647                                    .ENDM   CWD2
1648
1649                                    .MACRO  STDER1  B
1650                                            F.EAB   <D.DRA!B>
1651                                            CHECK   165,166,167,170,<AFTER TIMEOUT>,0,0,0
1652                                    .ENDM   STDER1
1653
1654                                    .MACRO  STDER2  B
1655                                            F.EAB   <D.DRA!B>
1656                                            CHECK   211,212,213,214,<AFTER RELEASE ISSUED>,0,0,0
1657                                    .ENDM   STDER2
1658
1659
1660                                    .MACRO  DRCLR   C,?A
1661
1662                                            MOV     #CCLR,RKCS1(R5)
1663                                            MOV     $UNIT,RKCS2(R5) ;DRIVE#
1664                                            ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
1665                                            MOV     #CLEAR,HCS1
1666                                            JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
1667                                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
1668                                            JSR     PC,TSTATN       ;TEST FOR ATTN
1669                                            BR      A
1670                                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1671                                    A:
1672                                    .IF     B       C
1673                                            F.EAB   D.DRA
1674                                            CHECK   33,34,35,36,<AFTER DRV CLEAR CMD>,T.A2,T.B2,0
1675                                    .ENDC           .
1676
1677                                    .ENDM   DRCLR
1678
1679
1680                                    ;
1681                                    ;USE CALIB      X   TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
1682                                    ;
1683                                    .MACRO  CALIB   A,?C
1684
1685                                            MOV     #CCLR,RKCS1(R5)
1686                                            MOV     $UNIT,RKCS2(R5)
1687                                            ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
1688                                            MOV     #RECAL,HCS1
1689                                            JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
1690                                            ERROR   124             ;RDY NOT SET AFTER RECAL CMD
1691
1692                                            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
1693                                            JSR     PC,GSTAT
1694                                            BIT     #D.RTZ,HMR2
```

```
                BNE     C
                ERROR   70              ;RTZ NOT SET DURING RECAL CMD
C:              MOV     T10,TEMP2       ;SETUP TIMEOUT
                JSR     PC,FATT1        ;FIND ATTN
                ERROR   55              ;NO ATTN AFTER RECAL CMD
.IF B           A
                F.EAB   <D.DSC!D.DRA>
                CHECK   221,66,222,67,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
                CWD2    47,50,<AFTER RECAL CMD>
.ENDC
                DRCLR

.ENDM           CALIB

;
;QUICK START SPINDLE
;
.MACRO          QKSRT

                JSR     PC,SUBCLR
                ERROR   24              ;CERR AFTER SCLR

                MOV     #SRTSPL,HCS1
                JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
                ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.

                MOV     T100,TEMP2      ;SETUP TIMEOUT
                JSR     PC,FATT1        ;FIND ATTN
                ERROR   74              ;NO ATTN AFTER ST SPIN CMD.

                CLR     UNLD

.ENDM           QKSRT

;
; A=WRHEAD/<CFMT!WRHEAD>
; USE WRHDR     <A>,X   TO OMIT CHECKING A0, B0, A1 & B1
;
.MACRO          WRHDR   A,C,?D

                MOV     #<A>,HCS1
                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
                ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
                JSR     PC,GSTAT        ;GET FRESH STATUS
                BIT     #CERR,HCS1
                BEQ     D
                ERROR   201             ;CERR AFTER WRITE HEADER CMD
                TYPE    ,MSG21          ;ABORTING BALANCE OF TESTS
                JMP     SEOP
D:
.IF B           C
                F.EAB   D.DRA
                CHECK   37,40,41,42,<AFTER WRITE HEADER CMD>,0,0,0
.ENDC

.ENDM           WRHDR
```

```
1751
1752                                    ;
1753                                    ; A=RDHEAD/<CFMT!RDHEAD>
1754                                    ;USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
1755                                    ;
1756                                    .MACRO  RDHDR   A,C,?D,?E
1757
1758                                            MOV     #RHTAB,R0
1759                                            MOV     #<A>,HCS1
1760                                            JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1761                                            ERROR   171             ;NO RDY AFTER READ HEADER CMD
1762                                            BIT     #CERR,HCS1
1763                                            BEQ     D
1764                                            ERROR   174             ;CERR AFTER READ HEADER CMD
1765
1766                                    D:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
1767                                            MOV     RKDB(R5),(R0)+  ;2'ND WORD
1768                                            MOV     RKDB(R5),(R0)+  ;3'RD WORD
1769
1770
1771                                            BIT     #DLT,RKCS2(R5)
1772                                            BEQ     E
1773                                            JSR     PC,GSTAT
1774                                            ERROR   173             ;DLT AFTER READ HEADER CMD
1775                                    E:
1776                                    .IF     B       C
1777                                            F.EAB   D.DRA
1778                                            CHECK 301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
1779                                    .ENDC
1780
1781                                    .ENDM   RDHDR
1782
1783
1784                                    .MACRO  HDCHK3  ?A
1785
1786                                            RDHDR   RDHEAD
1787                                            CMP     RHTAB,TOCYL     ;CHECK WORD 0 ONLY, CYL#
1788                                            BEQ     A               ;BR IF SAME
1789                                            ERROR   51              ;WRONG CYL# ON HEADER
1790                                    A:
1791
1792                                    .ENDM   HDCHK3
1793
1794                                    ;
1795                                    ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1796                                    ;
1797                                    .MACRO  HDTBL   A,B,C
1798
1799                                            MOV     A,CALADD        ;SETUP
1800                                            MOV     #B,HEAD         ;TO FILL
1801                                            MOV     #C,FORMAT       ;HEADER
1802                                            JSR     PC,FHDTAB       ;TABLE
1803
1804                                    .ENDM   HDTBL
1805
1806                                    ;
```

```
                        ;QUICK SEEK.    ENTER WITH CYL# IN RKDC
                        ;
                        .MACRO  QKSEEK  ?A

                                MOV     #SEEK,HCS1
                                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
                                ERROR   131             ;NO RDY AFTER SEEK CMD

                                MOV     T50000,TEMP1    ;SETUP TIMEOUT
                                JSR     PC,FATT2        ;FIND ATTN
                                ERROR   132             ;NO ATTN AFTER SEEK CMD

                                BIT     #CERR,HCS1
                                BEQ     A
                                ERROR   210             ;CERR AFTER SEEK CMD

                        A:      F.EAB   <D.DSC!D.DRA>
                                CHECK   161,162,163,164,<AFTER SEEK CMD>,0,0,0

                        .ENDM   QKSEEK


                        ;
                        ;A=WRDATA/<CFMT!WRDATA>
                        ;C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
                        ;D=ADDR TO JMP TO BYPASS TEST
                        ;E:  IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
                        ;E:  IF NON BLANK WILL OMIT CHECKING A0 THRU B1
                        ;

                        .MACRO  WDATA   A,C,D,E,?F,?G,?H,?I

                                MOV     #<A>,HCS1
                                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
                                ERROR   11              ;NO RDY AFTER WRITE DATA CMD
                                JSR     PC,GSTAT        ;GET FRESH STATUS
                                BIT     #CERR,HCS1
                                BEQ     I               ;BR IF NO ERRORS

                                BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
                                BEQ     G               ;BR IF NO
                                JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
                                BR      H               ;RETURN HERE IF NO

                                INC     SECTOR          ;RETURN HERE IF YES
                                CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
                                BNE     F               ;BR IF NO
                                ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
                                JMP     D               ;BYPASS TEST
                        F.      MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
                                JMP     C
                        G:      ERROR   12              ;CERR WITH WRITE DATA CMD
                                F.EAB   D.DRA
                                CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
                                TYPE    ,MSG21          ;ABORTING BALANCE OF TESTS
                                JMP     $EOP
```

```
1863                                    H:      ERROR   43              ;BAD SECTOR NOT LISTED IN TABLE
1864                                    I:
1865                                    .IF     B       E
1866                                            F.EAB   D.DRA
1867                                            CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1868                                    .ENDC
1869                                    .ENDM   WDATA
1870
1871
1872                                    ;
1873                                    ;A=RDDATA/<CFMT!RDDATA>
1874                                    ;USE RDATA      <A>,X TO OMIT CHECKING A0, B0, A1 & B1
1875                                    ;
1876
1877                                    .MACRO  RDATA   A,C,?D,?E,?F,?G,?H
1878
1879                                            MOV     #<A>,HCS1
1880                                            JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1881                                            ERROR   13              ;NO RDY AFTER READ DATA CMD
1882                                            JSR     PC,GSTAT        ;GET FRESH STATUS
1883                                            BIT     #CERR,HCS1
1884                                            BEQ     G
1885                                            BIT     #BSE,HER        ;SEE IF BAD SECTOR
1886                                            BEQ     E
1887                                            ERROR   65              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1888                                            BR      H
1889
1890                                    E:      BIT     #DCK,HER        ;SEE IF DATA CHECK ERROR
1891                                            BEQ     F
1892                                            ERROR   21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
1893                                            BR      H
1894
1895                                    F:      ERROR   14              ;CERR AFTER READ DATA CMD.
1896
1897                                    H:      F.EAB   D.DRA
1898                                            CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1899                                            TYPE    ,MSG21          ;ABORTING BAL OF TESTS
1900                                            JMP     $EOP
1901                                    G:
1902                                    .IF     B       C
1903                                            F.EAB   D.DRA
1904                                            CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1905                                    .ENDC
1906                                    .ENDM   RDATA
1907
1908
1909                                    ;
1910                                    ;A=WRTCHK/<CFMT!WRTCHK>
1911                                    ;C=EXPECTED DATA FOR TYPEOUT
1912                                    ;USE WRCHK      <A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
1913                                    ;
1914
1915                                    .MACRO  WRCHK   A,C,D,?E,?F
1916
1917                                            MOV     #<A>,HCS1
1918                                            JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
```

```
1919                                    ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
1920                                    JSR     PC,GSTAT        ;GET FRESH STATUS
1921                                    BIT     #CERR,HCS1
1922                                    BEQ     F
1923                                    BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
1924                                    BEQ     E
1925                                    MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
1926                                    MOV     C,WD2           ;EXPECTED WORD FOR TYPEOUT
1927                                    ERROR   16              ;WCE AFTER WRITE CMD
1928                                    BR      F
1929
1930                            E:      ERROR   22              ;CERR AFTER WRITE CHECK CMD
1931                                    F.EAB   D.DRA
1932                                    CHECK   57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1933                                    TYPE    ,MSG21          ;ABORTING BALANCE OF TESTS
1934                                    JMP     $EOP
1935
1936                            F:
1937                            .IF     B       D
1938                                    F.EAB   D.DRA
1939                                    CHECK   57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1940                            .ENDC
1941                            .ENDM   WRCHK
1942
1943
1944
1945                            .MACRO  EOPGM
1946
1947                                    SCOPE
1948                                    CLR     $ESCAPE
1949                                    MOV     #1,$TIMES
1950                                    MOV     #STACK,SP
1951                                    INC     $DEVCT          ;INCR COUNT FOR # DRIVES CHECKED
1952                                    CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
1953                                    BEQ     $EOP1+2         ;BR IF YES
1954                                    JMP     NUDRV           ;ELSE TEST NEXT DRIVE PRESENT
1955                            $EOP1:  SCOPE
1956                            .ENDM   EOPGM
1957
1958
1959                            .MACRO  QKPACK  ?A
1960                                    MOV     #CCLR,RKCS1(R5)
1961                                    MOV     $UNIT,RKCS2(R5)
1962                                    ADD     UNITB,RKCS2(R5)
1963                                    MOV     #PACK,HCS1
1964                                    JSR     PC,DOCMD        ;DO PACK CMD & GET CONTR RDY
1965                                    ERROR   116             ;CONTR NOT RDY AFT PACK CMD
1966
1967                                    BIT     #D.VV,HMR2
1968                                    BNE     A
1969                                    ERROR   27              ;VV NOT SET AFTER PACK CMD
1970                            A:
1971                            .ENDM   QKPACK
1972
1973
1974                            ;
```

```
1975                            ;MACRO TO RELEASE PORT A OR B
1976                            ;RLS PORT A:   RELEAS 0,1,A,B  (0,B,C,D IF USED WITH MACROS BELOW)
1977                            ;RLS PORT B:   RELEAS 1,0,B,A  (B,A,D,C IF USED WITH MACROS BELOW)
1978                            ;
1979                            .MACRO  RELEAS  W,X,Y,Z,?J
1980                                    MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT Y
1981                                    MOV     #W,UNITB
1982                                    ADD     UNITB,RKCS2(R5)
1983                                    MOVB    #''Y,MSG19A
1984                                    ADD     #RLS,RKCS2(R5)  ;RELEASE PORT Y
1985                                    MOV     #SELDRV,HCS1
1986                                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
1987                                    ERROR   117             ;NO RDY AFTER SEL DRV CMD
1988
1989                                    MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT Z
1990                                    MOV     #X,UNITB
1991                                    ADD     UNITB,RKCS2(R5)
1992                                    MOVB    #''Z,MSG19A
1993                                    MOV     #SELDRV,HCS1
1994                                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
1995                                    ERROR   117             ;NO RKY AFTER SEL DRV CMD
1996
1997                                    BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT Z
1998                                    BNE     J               ;BR IF YES
1999                                    ERROR   71              ;PORT Z NOT AVAIL AFTER PORT Y RLS
2000                            J:
2001                            .ENDM   RELEAS
2002
2003
2004
2005
2006                        MACRO   SETP    A,C
2007                                MOV     #A,UNITB        ;SETUP PORT C
2008                                MOVB    #''C,MSG19A
2009                                MOV     TIMER,COUNT
2010                                JSR     PC,TMO          ;DO TIMEOUT
2011
2012                                JSR     PC,SUBCLR
2013                                ERROR   24              ;CERR AFTER SCLR
```

```
2014
2015                                    JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
2016                                    ERROR   45              ;PORT C NOT AVAIL AFTER TMO
2017                            .ENDM   SETP
2018
2019                            .MACRO  REPTST
2020                            ;*
2021                            ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
2022                            ;*
2023                            .ENDM   REPTST
2024
2025                            .MACRO  REPTA
2026                            ;*
2027                            ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B',
2028                            ;*      BUT THE SEEK IS TO CYLINDER 0
2029                            ;*
2030                            .ENDM   REPTA
2031
2032                            .MACRO  HEADER
2033                            ;
2034                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2035                            ;C & D=A FOR PORT A OR B FOR PORT B
2036                            ;
2037                            .ENDM   HEADER
2038
2039                            .MACRO  SUNIT
2040                                    MOV     $UNIT,RKCS2(R5)
2041                                    ADD     UNITB,RKCS2(R5)
2042                            .ENDM   SUNIT
2043
2044                            ;
2045                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2046                            ;C & D=A FOR PORT A OR B FOR PORT B
2047                            ;
2048                            .MACRO  TST5    A,B,C,D,E
2049                                    SETP    A,C
2050                                    MOV     #5$,$ESCAPE
2051                                    F.EAB   D.DRA
2052                                    CHECK   77,100,101,102,<AFTER TIMEOUT>,0,0,0
2053                                    MOV     #B,UNITB        ;SELECT PORT D BEFORE TIMEOUT OR RELEASE
2054                                    MOVB    #''D,MSG19A     ;SETUP ERROR MSG FOR PORT D
2055                                    CLR     $ESCAPE
2056                                    JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
2057                                    BR      1$              ;BR IF NOT AVAIL
2058                                    ERROR   103             ;PORT D AVAIL BEFORE TMO OR RELEASE
2059
2060                            1$:     BIT     #CERR,HCS1
2061                                    BNE     6$
2062                                    ERROR   130             ;CERR NOT SET AFTER SEL DRIVE & DRIVE NOT AVAIL
2063                            6$:     MOV     #<D.SPIN!D.VV>,E.A0
2064                                    MOV     E.A0,E.A1       ;MSG A & B SHOULD ALWAYS RETURN SAME
2065                                    BIS     E.DDT,E.A1
2066                                    CLR     E.B0            ;WORD 0 FOR PORT D
2067                                    CLR     E.B1
2068                                    MOV     #5$,$ESCAPE
2069                                    CHECK   104,105,106,107,<WHILE PORT D UNAVAILABLE>,0,0,0
```

```
2070
2071                            TST     $PASS
2072                            BEQ     8$              ;BR IF FIRST PASS
2073                            JMP     5$              ;ELSE EXIT TEST
2074                    8$:     CLR     $ESCAPE
2075                            MOV     #CCLR,RKCS1(R5)
2076                            MOV     #A,UNITB        ;SETUP FOR PORT C AGAIN
2077                            MOVB    #''C,MSG19A
2078                            MOV     #360,COUNT      ;SETUP 4 SEC TIMEOUT
2079                            JSR     PC,CLKON        ;TURN ON CLOCK
2080
2081                            JSR     PC,DRAV
2082                            ERROR   45              ;PORT C NOT AVAIL AFTER TIMEOUT
2083
2084                            MOV     #B,UNITB        ;SELECT PORT D BEFORE TIMEOUT OR RELEASE
2085                            MOVB    #''D,MSG19A
2086                            JSR     PC,DRAV         ;SEE IF PORT D DRIVE AVAIL
2087                            BR      3$              ;BR IF NOT AVAIL
2088                            ERROR   103             ;PORT D AVAIL BEFORE TMO OR RELEASE
2089
2090                    3$:     MOV     #CCLR,RKCS1(R5)
2091                            MOV     $UNIT,R4
2092                            ADD     UNITB,R4
2093                            JSR     PC.FATT3
2094                            ERROR   110             ;NO ATTN ON PORT D TO ALLOW SEIZE
2095
2096                            JSR     PC,CLKOF
2097                            JSR     PC,DRAV         ;SEE IF PORT D DRIVE AVAIL
2098                            ERROR   45              ;PORT D NOT AVAIL
2099
2100                            MOV     #5$,$ESCAPE
2101                            F.EAB   <D.DSC!D.DRA>
2102                            CHECK   77,100,101,102,<AFTER TIMEOUT>,0,0,0
2103                            CLR     $ESCAPE
2104                            MOV     #A,UNITB        ;SETUP FOR PORT C
2105                            MOVB    #''C,MSG19A
2106                            JSR     PC,TSTATN       ;TEST FOR ATTN ON PORT C
2107                            BR      4$
2108                            ERROR   111             ;PORT C ATTN SET W/O REQUEST PENDING
2109
2110                    4$:     MOV     #5$,$ESCAPE
2111                            MOV     #B,UNITB        ;SETUP FOR PORT D
2112                            MOVB    #''D,MSG19A
2113                            DRCLR
2114
2115                            MOV     #360,R1
2116                            SUB     COUNT,R1        ;R1-COUNT=R1
2117                            MULT    #17.,R1         ;MULT BY 16.66 MS
2118                            TYPE    ,E              ;PORT TIMEOUT
2119                            MOV     R1,-(SP)        ;PUSH BINARY ONTO STACK
2120                            JSR     PC,$SB2D        ;CONVERT TO ASCII
2121                            JSR     PC,$SUPRS       ;TYPE IT
2122                            TYPE    ,MSG22          ;MS
2123
2124                    5$:     CLR     $ESCAPE
2125                            JSR     PC,CLKOF
```

```
2126                            .ENDM   TST5
2127
2128                            ;
2129                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2130                            ;C & D=A FOR PORT A OR B FOR PORT B
2131                            ;
2132                            .MACRO  TST10   A,B,C,D,E,F,?G
2133                                    SETP    A,C
2134                                    MOV     #F,RKDC(R5)     ;SEEK TO CYL F
2135                                    QKSEEK
2136                                    JSR     PC,RDCYLA       ;READ CYL ADDR
2137                                    CMP     CYLADD,#F       ;SEE IF CYL F
2138                                    BEQ     G               ;BR IF YES
2139                                    MOV     #E,FRCYL
2140                                    MOV     #F,TOCYL
2141                                    MOV     #F,CALDIF
2142                                    MOV     #F,RKDC(R5)     ;REFRESH RKDC
2143                                    ERROR   224             ;DID NOT SEEK TO CYL F
2144
2145                            G:      MOV     #150.,COUNT
2146                                    JSR     PC,TMO          ;DO 2.5 SEC TIMEOUT
2147                                    JSR     PC,FATT2
2148                                    ERROR   112             ;ATTN CLEARED AFTER TMO
2149                                    DRCLR
2150                                    JSR     PC,DRAV
2151                                    ERROR   113             ;PORT C NOT AVAIL AFTER DRIVE CLEAR CMD
2152                            .ENDM   TST10
2153
2154                            ;
2155                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2156                            ;C & D=A FOR PORT A OR B FOR PORT B
2157                            ;
2158                            .MACRO  TST12   A,B,C,D
2159                                    SETP    A,C
2160                                    QKSEEK
2161                                    MOV     TIMER,COUNT
2162                                    JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT
2163                                    MOV     #B,UNITB        ;SETUP PORT D
2164                                    MOVB    #''D,MSG19A
2165                                    JSR     PC,DRAV
2166                                    ERROR   45              ;PORT D NOT AVAIL AFTER TMO
2167
2168                                    STDER1  0
2169                                    QKSEEK
2170                                    MOV     TIMER,COUNT
2171                                    JSR     PC,TMO
2172                                    MOV     #A,UNITB        ;SETUP PORTC
2173                                    MOVB    #''C,MSG19A
2174                                    JSR     PC,DRAV
2175                                    ERROR   45              ;PORT C NOT AVAIL AFTER TMO
2176
2177                                    STDER1  D.DSC
2178                                    DRCLR
2179                                    MOV     TIMER,COUNT
2180                                    JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT
2181                                    MOV     #B,UNITB        ;SETUP PORT D
```

```
2182                                    MOVB    #''D,MSG19A
2183                                    JSR     PC,TSTATN
2184                                    ERROR   114             ;ATTN RESET ON PORT D AFTER DR CLR CMD
2185                                                            ;ON PORT C
2186
2187                                    STDER1  D.DSC
2188                            .ENDM   TST12
2189
2190                            ;
2191                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2192                            ,C & D=A FOR PORT A OR B FOR PORT B
2193                            ;
2194                            .MACRO  TST14   A,B,C,D
2195                                    SETP    A,C
2196                                    RELEAS  A,B,C,D
2197                                    JSR     PC,TSTATN
2198                                    BR      1$
2199                                    ERROR   115             ;ATTN SET IN PORT D AFTER RLS OF PORT C
2200                            1$:     F.EAB   D.DRA
2201                                    CHECK   133,134,135,136,<AFTER RELEASED TO PORT D>,0,0,0
2202
2203                                    MOV     #A,UNITB        ;ADDRESS PORT C
2204                                    MOVB    #''C,MSG19A
2205                                    JSR     PC.TSTATN
2206                                    SKIP    R,<GOTO NEXT TST>
2207                                    ERROR   115             ;ATTN SET AFTER RLS ISSUED
2208                                    STDER2  D.DRA
2209                            .ENDM   TST14
2210
2211                            ;
2212                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2213                            ;C & D=A FOR PORT A OR B FOR PORT B
2214                            ;
2215                            .MACRO  TST16   A,B,C,D
2216                                    RELEAS  B,A,D,C
2217                                    STDER2  D.DRA
2218                                    JSR     PC,SUBCLR
2219                                    ERROR   24              ;CERR AFTER SCLR
2220                                    QKSEEK
2221
2222                                    MOV     #B,UNITB        ;SETUP PORT D
2223                                    MOVB    #''D,MSG19A
2224                                    JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
2225                                    BR      1$
2226                                    ERROR   103             ;PORT D AVAIL BEFORE TMO OR RELEASE
2227
2228                            1$:     BIT     #CERR,HCS1
2229                                    BNE     2$
2230                                    ERROR   130             ;CERR NOT SET AFTER SEL DRIVE CMD
2231                                                            ;& NO DRA
2232                            2$:     JSR     PC,TSTATN
2233                                    BR      3$
2234                                    ERROR   115             ;ATTN SET IN PORT D AFTER RLS FROM PORT C
2235
2236                            3$:     MOV     #CCLR,RKCS1(R5)
2237                                    RELEAS  A,B,C,D
```

```
2238                                    JSR     PC,TSTATN
2239                                    ERROR   122             ;NO ATTN IN PORT D AFTER RLS FROM PORT C
2240
2241                                    STDER2  <D.DSC!D.DRA>
2242                                    RELEAS  B,A,D,C
2243                                    JSR     PC,TSTATN
2244                                    ERROR   123             ;ATTN CLEARED IN PORT C AFT RLS FROM PORT D
2245
2246                                    STDER2  <D.DSC!D.DRA>
2247                                    RELEAS  A,B,C,D
2248                                    DRCLR
2249                                    MOV     TIMER,COUNT
2250                                    JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT ON PORT D
2251                                    JSR     PC,TSTATN
2252                                    SKIP    R,<GOTO NEXT TST>
2253                                    ERROR   144             ;MULT ATTN ON PORT D
2254                            .ENDM   TST16
2255
2256
2257                            ;
2258                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2259                            ;C & D=A FOR PORT A OR B FOR PORT B
2260                            ;
2261                            .MACRO  TST20   A,B,C,D
2262                                    RELEAS  B,A,D,C
2263                                    STDER2  D.DRA
2264                                    JSR     PC,SUBCLR
2265                                    ERROR   24              ;CERR AFTER SCLR
2266
2267                                    MOV     #B,UNITB        ;SETUP PORT D
2268                                    MOVB    #''D,MSG19A
2269                                    JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
2270                                    BR      1$
2271                                    ERROR   103             ;PORT D AVAIL BEFORE TMO OR RELEASE
2272
2273                            1$:     BIT     #CERR,HCS1
2274                                    BNE     2$
2275                                    ERROR   130             ;CERR NOT SET BY NO DRA
2276
2277                            2$:     JSR     PC,TSTATN
2278                                    BR      3$
2279                                    ERROR   115             ;ATTN SET IN PORT D AFTER RLS FROM PORT C
2280
2281                            3$:     MOV     #CCLR,RKCS1(R5)
2282                                    RELEAS  B,A,D,C
2283                                    RELEAS  A,B,C,D
2284                                    MOV     TIMER,COUNT
2285                                    JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT ON PORT D
2286                                    JSR     PC,TSTATN
2287                                    BR      4$
2288                                    ERROR   115             ;ATTN SET ON PORT D AFTER RLS FROM PORT D
2289
2290                            4$:     STDER2  0
2291                                    MOV     #A,UNITB        ;SETUP PORT C
2292                                    MOVB    #''C,MSG19A
2293                                    JSR     PC,TSTATN
```

```
2294                                    SKIP    R,<GOTO NEXT TST>
2295                                    ERROR   115             ;ATTN SET ON PORT C, AFTER RLS FROM PORT C
2296                                .ENDM   TST20
2297
2298
2299                                ;
2300                                ;A & B=0 FOR PORT A OR 1 FOR PORT B
2301                                ;C & D=A FOR PORT A OR B FOR PORT B
2302                                ;
2303                                .MACRO  TST22   A,B,C,D
2304                                        SETP    A,C
2305                                        RELEAS  B,A,D,C
2306                                        STDER2  D.DRA
2307                                .ENDM   TST22
2308
2309                                ;
2310                                ;A & B=0 FOR PORT A OR 1 FOR PORT B
2311                                ;C & D=A FOR PORT A OR B FOR PORT B
2312                                ;
2313                                .MACRO  TST24   A,B,C,D,E
2314                                        RELEAS  B.A,D,C
2315                                        JSR     PC,SUBCLR
2316                                        ERROR   24              ;CERR AFTER SCLR
2317
2318                                        MOV     #E,PKDC(R5)
2319                                        MOV     #SEEK,HCS1
2320                                        JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
2321                                        ERROR   131             ;NO RDY AFTER SEEK CMD
2322                                        RELEAS  A,B,C,D
2323                                        MOV     T50000,TEMP1
2324                                        JSR     PC,FATT2
2325                                        ERROR   152             ;NO ATTN ON PORT D AFTER SEEK & RLS FROM PORT C
2326
2327                                        F.EAB   <D.DRA!D.DSC>
2328                                        CHECK   145,146,147,150,<AFTER SEEK & RLS FROM PORT C>,0,0,0
2329                                        RELEAS  B.A,D,C
2330                                        JSR     PC,TSTATN
2331                                        BR      1$
2332                                        ERROR   160             ;ATTN ON PORT C AFTER SEEK & RLS FROM PORT C
2333
2334                                1$:     F.EAB   D.DRA
2335                                        CHECK   145,146,147,150,<AFTER SEEK & RLS FROM PORT C>,0,0,0
2336                                .ENDM   TST24
2337
2338                                ;
2339                                ;A & B=0 FOR PORT A OR 1 FOR PORT B
2340                                ;C & D=A FOR PORT A OR B FOR PORT B
2341                                ;
2342                                .MACRO  TST26   A,B,C,D
2343                                        JSR     PC,SUBCLR
2344                                        ERROR   24              ;CERR AFTER SCLR
2345                                        RELEAS  B,A,D,C
2346                                        JSR     PC,SUBCLR
2347                                        ERROR   24              ;CERR AFTER SCLR
2348
2349                                        MOV     #30.,COUNT
```

```
2350                                    JSR     PC,TMO          ;DO 500MS TIMEOUT
2351                                    JSR     PC,DRAV         ;RE-SEIZE DRIVE THRU PORT C
2352                                    ERROR   45              ;PORT C NOT AVAIL AFTER TIMEOUT
2353                                    STDER1  0
2354                                    MOV     #-1,COUNT
2355                                    JSR     PC,CLKON        ;TURN ON CLOCK
2356                                    MOV     #B,UNITB        ;SETUP PORT D
2357                                    JSR     PC,DRAV         ;SEE IF DRV AVAIL
2358                                    BR      1$              ;BR IF NO
2359                                    ERROR   103             ;PORT C AVAIL BEFORE TMO OR RLS
2360                            1$:     MOVB    #''D,MSG19A
2361                                    MOV     $UNIT,R4
2362                                    ADD     UNITB,R4
2363                                    JSR     PC,FATT3
2364                                    ERROR   110             ;NO ATTN ON PORT D TO ALLOW SEIZE
2365
2366                                    JSR     PC,CLKOF        ;TURN CLOCK OFF
2367                                    COM     COUNT           ;GET ACTUAL COUNT OF TIMEOUT
2368                                    CMP     COUNT,#35.      ;COMPARE COUNT AGAINST APPROX 1 SEC
2369                                    SKIP    GE,<GO TO NEXT TEST>
2370                                    ERROR   153             ;TIMEOUT DID NOT RE-TRIGGER
2371                            .ENDM   TST26
2372
2373
2374                            ;
2375                            ;A & B=0 FOR PORT A OR 1 FOR PORT B
2376                            ;C & D=A FOR PORT A OR B FOR PORT B
2377                            ;
2378                            .MACRO  TST30   A,B,C,D
2379                                    JSR     PC,SUBCLR
2380                                    ERROR   24              ;CERR AFTER SCLR
2381                                    RELEAS  B,A,D,C
2382                                    JSR     PC,SUBCLR
2383                                    ERROR   24              ;CERR AFTER SCLR
2384
2385                                    MOV     #B,UNITB        ;SETUP FOR PORT D
2386                                    MOVB    #''D,MSG19A
2387                                    JSR     PC,DRAV         ;PORT D TRIES TO SEIZE THE DRIVE
2388                                    BR      1$              ;BR IF NOT AVAIL
2389                                    ERROR   103             ;PORT D AVAIL BEFORE TMO OR RELEASE
2390
2391                            1$:     MOV     #CCLR,RKCS1(R5)
2392                                    RELEAS  A,B,C,D
2393                                    MOV     #A,UNITB        ;SETUP FOR PORT C
2394                                    MOVB    #''C,MSG19A
2395                                    JSR     PC,DRAV
2396                                    BR      2$
2397                                    ERROR   103             ;PORT C AVAIL BEFORE TMO OR RELEASE
2398
2399                            2$:     MOV     #CCLR,RKCS1(R5)
2400                                    MOV     #-1,COUNT
2401                                    JSR     PC,CLKON
2402                                    MOV     $UNIT,R4
2403                                    ADD     UNITB,R4
2404                                    JSR     PC,FATT3
2405                                    ERROR   110             ;NO ATTN ON PORT C TO ALLOW SEIZE
```

```

                JSR     PC,CLKOF                ;TURN CLOCK OFF
                COM     COUNT                   ;GET ACTUAL COUNT OF TIMEOUT
                CMP     COUNT,#35.              ;COMPARE AGAINST APPROX 1 SEC
                SKIP    GE,<GO TO NEXT TST>
                ERROR   153                     ;TIMEOUT DID NOT RE-TRIGGER
        .ENDM   TST30



;
;A & B=0 FOR PORT A OR 1 FOR PORT B
;C & D=A FOR PORT A OR B FOR PORT B
;
        .MACRO  TST32   A,B,C,D
                RELEAS  B,A,D,C
                JSR     PC,SUBCLR
                ERROR   24                      ;CERR AFTER SCLR

                INC     UNLD                    ;USED FOR VALID HALT
                MOV     #2$,$ESCAPE
                MOV     #UNLOAD,HCS1            ;UNLOAD CMD
                BIS     $TMP4,HCS1
                MOV     HCS1,RKCS1(R5)
                MOV     T10,TEMP1               ;SETUP TIMEOUT
                JSR     PC,FATT2                ;FIND ATTN
                ERROR   73                      ;NO ATTN AFTER UNLD CMD

                MOV     #300.,COUNT
                JSR     PC,TMO                  ;DO 5 SEC DELAY
                MOV     #B,UNITB                ;SETUP FOR PORT D
                MOVB    #''D,MSG19A
                JSR     PC,DRAV
                BR      1$
                ERROR   155                     ;PORT D AVAIL BEFORE RLS WHEN UNLOADED
                                                ;UNLOAD DID NOT INHIBIT TIMERS

        1$:     MOV     #CCLR,RKCS1(R5)
                RELEAS  A,B,C,D
        2$:     CLR     $ESCAPE
                QKSRT
                CLR     UNLD
        .ENDM   TST32

;
;A & B=0 FOR PORT A OR 1 FOR PORT B
;C & D=A FOR PORT A OR B FOR PORT B
;
        .MACRO  TST32A  A,B,C,D
                RELEAS  B,A,D,C
                JSR     PC,SUBCLR
                ERROR   24                      ;CERR AFTER SCLR

                MOV     #5$,$ESCAPE
                MOV     LC,RKDC(R5)             ;SEEK TO LAST CYL
                QKSEEK
                DRCLR
```

```
2462                                    MOV     #RECAL,HCS1     ;RECAL COMMAND
2463                                    BIS     $TMP4,HCS1
2464                                    MOV     HCS1,RKCS1(R5)
2465                                    MOV     $UNIT,R4
2466                                    ADD     UNITB,R4
2467                            1$:     BITB    ATTN(R4),RKASOF+1(R5)   ;SEE IF ATTN SET
2468                                    BNE     3$              ;BR IF YES
2469
2470                                    MOV     #B,UNITB        ;SETUP FOR PORT D
2471                                    MOVB    #''D,MSG19A
2472                                    JSR     PC,DRAV         ;SEE IF DRV AVAIL
2473                                    BR      2$              ;RETURN HERE IF NO
2474                                    ERROR   177             ;PORT D AVAIL
2475                                                            ;RECAL DID NOT INHIBIT TIMERS
2476                            2$:     MOV     #CCLR,RKCS1(R5)
2477                                    MOV     #A,UNITB
2478                                    MOVB    #''C,MSG19A     ;SETUP PORT C
2479                                    JSR     PC,DRAV         ;SEE IF DRV AVAIL
2480                                    ERROR   203             ;PORT C NOT REMAIN AVAIL DURING RECAL
2481                                    BR      1$
2482
2483                            3$:     JSR     PC,GSTAT
2484                                    BIT     #D.PIP,HMR2     ;SEE IF ANY MOTION
2485                                    BEQ     4$              ;BR IF NO
2486                                    ERROR   72              ;PIP SET AFTER ATTN REC'D FROM RECAL
2487                            4$:     MOV     #120.,COUNT
2488                                    JSR     PC,TMO          ;DO 2 SEC DLY
2489                                    MOV     #B,UNITB        ;SETUP FOR PORT D
2490                                    MOVB    #''D,MSG19A
2491                                    JSR     PC,DRAV         ;SEE IF DRV NOW AVAIL
2492                                    ERROR   45              ;PORT D NOT AVAIL AFTER TMO
2493
2494                            5$:     CLR     $ESCAPE
2495                            .ENDM   TST32A
2496
2497
2498                                    .NLIST  MD
2499
```

```
                        .SBTTL  COMMON TAGS

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

                                .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG.  ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG.  ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001166  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001170  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001172  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001174  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001176  000000          $ESCAPE: 0                      ;;ESCAPE ON ERROR ADDRESS
001200  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001204  077               $QUES:  .ASCII  /?/           ;;QUESTION MARK
001205  015               $CRLF:  .ASCII  <15>          ;;CARRIAGE RETURN
001206  000012            $LF:    .ASCIZ  <12>          ;;LINE FEED
                        ;;**********************************************************************
                        .SBTTL  APT MAILBOX-ETABLE

                        ;;**********************************************************************
                        .EVEN
001210                  $MAIL:                          ;;APT MAILBOX
001210  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
```

```
2556  001212  000000                  $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
2557  001214  000000                  $TESTN: .WORD   ATESTN  ;;TEST NUMBER
2558  001216  000000                  $PASS:  .WORD   APASS   ;;PASS COUNT
2559  001220  000000                  $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
2560  001222  000000                  $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
2561  001224  000000                  $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
2562  001226  000000                  $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
2563  001230                          $ETABLE:                ;;APT ENVIRONMENT TABLE
2564  001230     000                  $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
2565  001231     000                  $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
2566  001232  000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
2567  001234  000000                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
2568  001236  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
2569                                  ;*                      BITS 15-11=CPU TYPE
2570                                  ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2571                                  ;*                              11/70=06,PDQ=07,Q=10
2572                                  ;*                      BIT 10=REAL TIME CLOCK
2573                                  ;*                      BIT  9=FLOATING POINT PROCESSOR
2574                                  ;*                      BIT  8=MEMORY MANAGEMENT
2575  001240     000                  $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S.  BYTE
2576  001241     000                  $MTYP1: .BYTE   AMTYP1  ;;MEM.  TYPE,BLK#1
2577                                  ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
2578                                  ;*                              900 NSEC CORE=001
2579                                  ;*                              300 NSEC BIPOLAR=002
2580                                  ;*                              500 NSEC MOS=003
2581  001242  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
2582                                  ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2583  001244     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S.  BYTE
2584  001245     000                  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
2585  001246  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
2586  001250     000                  $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
2587  001251     000                  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
2588  001252  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
2589  001254     000                  $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
2590  001255     000                  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
2591  001256  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
2592  001260  000000                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
2593  001262  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
2594  001264  177440                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
2595  001266  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
2596  001270  000000                  $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
2597  001272  000000                  $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
2598  001274  000000                  $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
2599  001276  000000                  $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
2600  001300  000000                  $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
2601  001302  000000                  $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
2602  001304  000000                  $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
2603  001306  000000                  $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
2604  001310  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
2605  001312  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
2606  001314                          $ETEND:
2607                                  .MEXIT
2608          177440                  ABASE=  177440                  ;DEFAULT BUSS ADDRESS
2609  001314  000210                  RKVEC:  210                     ;DEFAULT CONTROLLER INTERRUPT VECTOR
2610  001316  000240                  RKPRI:  PR5                     ;PRIORITY
2611  001320  172540                  PKS:    172540                  ;P-CLOCK STATUS REG
```

```
2612  001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
2613  001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
2614  001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.
2615
2616  001330  000100          LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
2617  001332  000104          PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.
2618
2619          000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
2620          172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
2621  001334  000000          TRAPPC: 0               ;PC FOR MEM CHECK ENABLE TRAP
2622
2623  001336  000000          PARAM:  0               ;1 FOR 220 START, NO DEFAULT
2624  001340  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
2625
2626  001342  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
2627                                                  ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
2628          000040          SPBAR=40                ;SPACEBAR
2629  001344  000000          FRCYL:  0               ;FROM CYLINDER
2630  001346  000000          TOCYL:  0               ;TO CYLINDER
2631  001350  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
2632  001352  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
2633  001354  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
2634  001356  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
2635  001360  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
2636  001362  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
2637
2638  001364  000074          HZ:     60.             ;60 FOR 60 CPS
2639                                                  ;50 FOR 50 CPS
2640  001366  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
2641                                                  ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
2642  001370  000000          SEC:    0               ;SECOND COUNTER
2643  001372  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
2644  001374  000000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
2645
2646  001376  000001          T1:     1               ;TIMEOUT CONSTANTS
2647  001400  000012          T10:    10.
2648  001402  000062          T50:    50.
2649  001404  000764          T500:   500.
2650  001406  000144          T100:   100.
2651  001410  011610          T5000:  5000.
2652  001412  141520          T50000: 50000.
2653
2654
2655  001414  000000          WD1:    0               ;ACTUAL HEADER/DATA WORD
2656  001416  000000          WD2:    0               ;EXPECTED DATA WORD
2657  001420  000000          HEAD:   0               ;HEAD NUMBER
2658  001422  000000          HEADA:  0               ;HEAD # FROM H.B3, RT. JUSTIFIED
2659  001424  000000          HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
2660  001426  000000          FORMAT: 0               ;FORMAT TYPE
2661  001430  000000          FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
2662  001432  000000          WDCNT:  0               ;WORD COUNT
2663
2664  001434  000000          DATA0:  0               ;ALL 0'S
2665  001436  052525          DATA01: 52525           ;0101 PATT
2666  001440  177777          DATA1:  177777          ;ALL 1'S
2667  001442  133467          DPAT1:  133467
```

```
2668  001444  070627                  DPAT2:  70627
2669
2670  001446  000000                  WORD:   0                       ;HEADER/DATA WORD
2671  001450  000000                  HDWD:   0                       ;HEADER WORD FROM RKDB
2672
2673  001452  000000                  BSERR:  0                       ;CANNOT READ BSE INFO WHEN SET
2674  001454  000000                  LIMERR: 0                       ;LIMIT DETECT ERROR FLAG
2675  001456  000000                  BYPCERR:0                       ;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
2676  001460  000000                  CHKFLG: 0                       ;WORDS TO BE TESTED
2677
2678  001462  000102                  HDTAB:  .BLKW 66.               ;CALCULATED HEADER WORD TABLE
2679  001666  000102                  RHTAB:  .BLKW 66.               ;FILLED AFTER READ HEADER CMD
2680  002072  000102                  SRTTAB: .BLKW 66.               ;ABOVE RHTAB SORTED STARTING FORM
2681                                                                  ;SECTOR 0 BY SORT ROUTINE
2682  002276  000400                  BSE22H: .BLKW 256.              ;22 SECTOR HARDWARE BSE INFO.
2683  003276  000400                  BSE22S: .BLKW 256.              ;22 SECTOR SOFTWARE BSE INFO.
2684  004276  000400                  RDTAB:  .BLKW 256.              ;FILLED AFTER READ DATA CMD
2685
2686  005276  000000                  UNLD:   0                       ;SET TO 0 IF HEADS ARE LOADED
2687                                                                  ;SET TO 1 IF HEADS UNLOADED
2688  005300  000000                  BADHDR: 0                       ;SET TO 0 IF FORMATTING OK
2689                                                                  ;SET TO 1 IF FORMATTING ALTERED
2690  005302  000000                  HPEND:  0                       ;SET TO 0 IF HALT NOT PENDING
2691                                                                  ;SET TO 1 IF HALT PENDING
2692
2693                                                                  ;THE ABOVE 3 FLAGS ARE USED
2694                                                                  ;BY 'STOP' ROUTINE TO BRING
2695                                                                  ;THE CPU TO A VALID HALT.
2696
2697
2698  005304    001     002     004   ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
2699  005307    010     020     040
2700  005312    100     200
2701                                          .EVEN
2702
2703                                  ;
2704                                  ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
2705                                  ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
2706                                  ;
2707
2708  005314  000000                  HCS1:   0                       ;HOLD RKCS1
2709  005316  000000                  HCS2:   0                       ;HOLD RKCS2
2710  005320  000000                  HWC:    0                       ;HOLD RKWC
2711  005322  000000                  HBA:    0                       ;ETC.
2712  005324  000000                  HDA:    0
2713  005326  000000                  HDS:    0
2714  005330  000000                  HER:    0
2715  005332  000000                  HASOF:  0
2716  005334  000000                  HDC:    0
2717  005336  000000                  HDB:    0
2718  005340  000000                  HMR1:   0
2719  005342  000000                  HMR2:   0
2720  005344  000000                  HMR3:   0
2721  005346  000000                  HPOS:   0
2722  005350  000000                  HPAT:   0
2723
```

```
2724
2725  005352  000000          TEMP1:  0                       ;TEMPORARY STORAGE.
2726  005354  000000          TEMP2:  0
2727  005356  000000          TEMP3:  0
2728  005360  000000          TEMP4:  0
2729  005362  000000          TEMP5:  0
2730                          ;
2731                          ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
2732                          ;
2733  005364  000000          H.A0:   0
2734  005366  000000          H.B0:   0
2735  005370  000000          H.A1:   0
2736  005372  000000          H.B1:   0
2737  005374  000000          H.A2:   0
2738  005376  000000          H.B2:   0
2739  005400  000000          H.A3:   0
2740  005402  000000          H.B3:   0
2741                          ;
2742                          ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
2743                          ;
2744  005404  000000          E.A0:   0
2745  005406  000000          E.B0:   0
2746  005410  000000          E.A1:   0
2747  005412  000000          E.B1:   0
2748  005414  000000          E.A2:   0
2749  005416  000000          E.B2:   0
2750  005420  000000          E.A3:   0
2751  005422  000000          E.B3:   0
2752                          ;
2753                          ;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.
2754                          ;
2755          000001          T.A2=BIT0                       ;TEST MSG A2 IF SET
2756          000002          T.B2=BIT1
2757          000004          T.B3=BIT2
2758                          ;
2759                          ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
2760                          ;
2761  005424  000000          DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
2762  005426  000000          DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
2763  005430  000000          ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
2764  005432  000000          PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
2765  005434  000000          DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
2766                          ;
2767                          ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
2768                          ;IS PRESENT AND IS TO BE TESTED.
2769                          ;ONLY DRIVES 0,2,4,6 CAN BE TESTED.
2770                          ;
2771  005436  000000          DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
2772  005440  000000          DRIV1:  0                       ;FOR DRIVE 1
2773  005442  000000          DRIV2:  0                       ;FOR DRIVE 2
2774  005444  000000          DRIV3:  0                       ;FOR DRIVE 3
2775  005446  000000          DRIV4:  0                       ;FOR DRIVE 4
2776  005450  000000          DRIV5:  0                       ;FOR DRIVE 5
2777  005452  000000          DRIV6:  0                       ;FOR DRIVE 6
2778  005454  000000          DRIV7:  0                       ;FOR DRIVE 7
2779
```

```
2780  005456  000000                LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
2781  005460  000000                PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
2782  005462  000000                SIZFLG: 0                       ;SET IF DEFAULT DO SIZING IN TEST 1
2783  005464  000000                UNITB:  0                       ;0 FOR PORT A:  DRIVES 0,2,4,6
2784                                                                ;1 FOR PORT B:  DRIVES 1,3,5,7
2785  005466  000132                TIMER:  90.                     ;TIMER FOR 1.5 SEC
```

```
                                .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


005470                          $ERRTB:

                                ;ERROR 1
005470  057127                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
005472  062556                          DH1
005474  065530                          DT1
005476  066332                          DF1

                                ;ERROR 2
005500  057346                          EM5                     ;DETECTED MDS
005502  062556                          DH1
005504  065530                          DT1
005506  066332                          DF1

                                ;ERROR 3
005510  057367                          EM6                     ;DETECTED UFE
005512  062556                          DH1
005514  065530                          DT1
005516  066332                          DF1

                                ;ERROR 4
005520  057410                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
005522  062556                          DH1
005524  065530                          DT1
005526  066332                          DF1
                                ;ERROR 5
005530  000000                          0
005532  000000                          0
005534  000000                          0
005536  000000                          0

                                ;ERROR 6
005540  057654                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
005542  062556                          DH1
005544  065530                          DT1
005546  066332                          DF1

                                ;ERROR 7
005550  057730                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
005552  062556                          DH1
005554  065530                          DT1
005556  066332                          DF1
```

```
2842                                    ;ERROR 10
2843  005560  060013                          EM11                    ;DRA & NED BOTH SET
2844  005562  062556                          DH1
2845  005564  065530                          DT1
2846  005566  066332                          DF1
2847                                    ;ERR 11
2848  005570  060057                          EM12                    ;NO RDY
2849  005572  064320                          DH27                    ;AFTER WRITE DATA CMD
2850  005574  065530                          DT1
2851  005576  066462                          DF10
2852                                    ,ERR 12
2853  005600  060453                          EM21                    ;CERR SET
2854  005602  064320                          DH27
2855  005604  065530                          DT1
2856  005606  066462                          DF10
2857                                    ;ERR 13
2858  005610  060057                          EM12                    ;NO RDY
2859  005612  064270                          DH26                    ;AFTER READ DATA CMD
2860  005614  065530                          DT1
2861  005616  066462                          DF10
2862                                    ;ERR 14
2863  005620  060453                          EM21                    ;CERR SET
2864  005622  064270                          DH26
2865  005624  065530                          DT1
2866  005626  066462                          DF10
2867                                    ;ERR 15
2868  005630  060057                          EM12                    ;NO RDY
2869  005632  064450                          DH32                    ;AFTER WRITE CHECK CMD
2870  005634  065530                          DT1
2871  005636  066462                          DF10
2872                                    ;ERR 16
2873  005640  062236                          EM80                    ;WRITE CHECK ERROR SET
2874  005642  064450                          DH32                    ;AFTER WRITE CHECK CMD
2875  005644  065642                          DT6
2876  005646  066352                          DF3
2877                                    ;ERR 17
2878  005650  060137                          EM14                    ;UNEXP MEM PARITY TRAP
2879  005652  063214                          DH8                     ;TEST #, TRAP PC
2880  005654  065570                          DT3
2881  005656  066346                          DF2
2882                                    ;ERR 20
2883  005660  062275                          EM82                    ;READ DATA NOT COMPARE WITH WRITE DATA
2884  005662  064270                          DH26                    ;AFTER READ DATA CMD
2885  005664  065642                          DT6
2886
2887  005666  066352                          DF3
2888                                    ;ERR 21
2889  005670  062347                          EM83                    ;DATA CHECK ERROR
2890  005672  064270                          DH26
2891  005674  065530                          DT1
2892
2893  005676  066462                          DF10
2894                                    ;ERR 22
2895  005700  060453                          EM21                    ;CERR SET
2896  005702  064450                          DH32                    ;AFTER WRITE CHECK CMD
2897  005704  065530                          DT1
```

```
2898  005706  066462                    DF10
2899                              ;ERR 23
2900  005710  060370                    EM18          ;MSG B0 ERROR
2901  005712  064320                    DH27          ;AFTER WRITE DATA CMD
2902  005714  066066                    DT13
2903  005716  066612                    DF21
2904                              ;ERROR 24
2905  005720  060453                    EM21          ;CERR SET
2906  005722  064070                    DH21          ;AFTER SCLR
2907  005724  065530                    DT1
2908  005726  066462                    DF10
2909                              ;ERR 25
2910  005730  060432                    EM20          ;MSG B1 ERROR
2911  005732  064320                    DH27
2912  005734  066066                    DT13
2913  005736  066612                    DF21
2914                              ;ERR 26
2915  005740  060370                    EM18
2916  005742  064270                    DH26          ;AFTER READ DATA CMD
2917  005744  066066                    DT13
2918  005746  066612                    DF21
2919
2920                              ;ERROR 27
2921  005750  060702                    EM24          ;VOL VALID NOT SET
2922  005752  064012                    DH19          ;AFTER PACK CMD
2923  005754  065530                    DT1
2924  005756  066462                    DF10
2925                              ;ERR 30
2926  005760  060432                    EM20          ;MSG B1 ERROR
2927  005762  064270                    DH26          ;AFTER READ DATA CMD.
2928  005764  066066                    DT13
2929  005766  066612                    DF21
2930                              ;ERR 31
2931  005770  060370                    EM18          ;MSG B0 ERROR
2932  005772  064450                    DH32          ;AFTER WRITE CHECK CMD
2933  005774  066066                    DT13
2934  005776  066612                    DF21
2935                              ;ERR 32
2936  006000  060432                    EM20          ;MSG B1 ERROR
2937  006002  064450                    DH32
2938  006004  066066                    DT13
2939  006006  066612                    DF21
2940                              ;ERR 33
2941  006010  060347                    EM17          ;A0 ERROR
2942  006012  064116                    DH22          ;AFTER DRIVE CLR CMD
2943  006014  066066                    DT13
2944  006016  066612                    DF21
2945                              ;ERR 34
2946  006020  060370                    EM18          ;B0 ERROR
2947  006022  064116                    DH22
2948  006024  066066                    DT13
2949  006026  066612                    DF21
2950                              ;ERR 35
2951  006030  060411                    EM19
2952  006032  064116                    DH22
2953  006034  066066                    DT13
```

```
2954  006036  066612                    DF21
2955                              ;ERR 36
2956  006040  060432                    EM20
2957  006042  064116                    DH22
2958  006044  066066                    DT13
2959  006046  066612                    DF21
2960                              ;ERR 37
2961  006050  060347                    EM17          ;AC ERROR
2962  006052  064736                    DH39          ;AFTER WRITE HEADER CMD
2963  006054  066066                    DT13
2964  006056  066612                    DF21
2965                              ;ERR 40
2966  006060  060370                    EM18
2967  006062  064736                    DH39
2968  006064  066066                    DT13
2969  006066  066612                    DF21
2970                              ;ERR 41
2971  006070  060411                    EM19
2972  006072  064736                    DH39
2973  006074  066066                    DT13
2974  006076  066612                    DF21
2975                              ;ERR 42
2976  006100  060432                    EM20
2977  006102  064736                    DH39
2978  006104  066066                    DT13
2979  006106  066612                    DF21
2980                              ;ERR 43
2981  006110  061007                    EM26          ;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
2982  006112  064320                    DH27          ;AFTER WRITE DATA CMD
2983  006114  065530                    DT1
2984  006116  066462                    DF10
2985                              ;ERR 44
2986  006120  060175                    EM15          ;WCE AT CYL 411, TRK 2, SEC 21
2987  006122  062556                    DH1
2988  006124  065530                    DT1
2989  006126  066376                    DF4
2990                              ;ERR 45
2991  006130  061402                    EM31          ;PORT  NOT AVAIL
2992  006132  062760                    DH5           ;AFTER TIMEOUT
2993  006134  065530                    DT1
2994  006136  066462                    DF10
2995                              ;ERR 46
2996  006140  060735                    EM25          ;DETECTED 10 BAD SECTORS
2997  006142  064320                    DH27          ;AFTER WRITE DATA CMD.
2998  006144  065530                    DT1
2999  006146  066462                    DF10
3000                              ;ERROR 47
3001  006150  061713                    EM39          ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3002  006152  063741                    DH17          ;AFTER RECAL CMD
3003  006154  066146                    DT14
3004  006156  066646                    DF22
3005                              ;ERROR 50
3006  006160  061762                    EM40          ;CYL ADDR IN RKMR3 NOT CLEARED
3007  006162  063741                    DH17          ;AFTER RECAL COMD
3008  006164  066146                    DT14
3009  006166  066646                    DF22
```

```
3010                                    ;ERR 51
3011  006170  062475                          EM93            ;WRONG CYL# IN HEADER WORD (MISPOSITION)
3012  006172  064245                          DH25            ;AFTER SEEK CMD
3013  006174  066022                          DT9
3014  006176  066566                          DF20
3015                                    ;ERR 52
3016  006200  060347                          EM17            ;MSG A0 ERROR
3017  006202  064320                          DH27            ;AFTER WRITE DATA CMD
3018  006204  066066                          DT13
3019  006206  066612                          DF21
3020                                    ;ERR 53
3021  006210  060411                          EM19            ;MSG A1 ERROR
3022  006212  064320                          DH27
3023  006214  066066                          DT13
3024  006216  066612                          DF21
3025                                    ;ERR 54
3026  006220  060347                          EM17            ;MSG A0 ERROR
3027  006222  064270                          DH26            ;AFTER READ DATA CMD
3028  006224  066066                          DT13
3029  006226  066612                          DF21
3030                                    ;ERROR 55
3031  006230  060115                          EM13            ;NO ATTN
3032  006232  063741                          DH17            ;AFTER RECAL CMD
3033  006234  065530                          DT1
3034  006236  066462                          DF10
3035                                    ;ERR 56
3036  006240  060411                          EM19            ;MSG A1 ERROR
3037  006242  064270                          DH26
3038  006244  066066                          DT13
3039  006246  066612                          DF21
3040                                    ;ERR 57
3041  006250  060347                          EM17            ;MSG A0 ERROR
3042  006252  064450                          DH32            ;AFTER WRITE CHECK CMD
3043  006254  066066                          DT13
3044  006256  066612                          DF21
3045                                    ;ERR 60
3046  006260  060411                          EM19            ;MSG A1 ERROR
3047  006262  064450                          DH32
3048  006264  066066                          DT13
3049  006266  066612                          DF21
3050                                    ;ERR 61
3051  006270  060347                          EM17            ;A0 ERROR
3052  006272  064371                          DH30            ;AFTER READ HEADER CMD
3053  006274  066066                          DT13
3054  006276  066612                          DF21
3055                                    ;ERR 62
3056  006300  060370                          EM18
3057  006302  064371                          DH30
3058  006304  066066                          DT13
3059  006306  066612                          DF21
3060                                    ;ERR 63
3061  006310  060411                          EM19
3062  006312  064371                          DH30
3063  006314  066066                          DT13
3064  006316  066612                          DF21
3065                                    ;ERR 64
```

```
3066  006320  060432                  EM20
3067  006322  064371                  DH30
3068  006324  066066                  DT13
3069  006326  066612                  DF21
3070                          ;ERR 65
3071  006330  061066                  EM27          ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
3072  006332  062556                  DH1
3073  006334  065530                  DT1
3074  006336  066332                  DF1
3075                          ;ERR 66
3076  006340  060370                  EM18          ;BO ERROR
3077  006342  063741                  DH17          ;AFTER RECAL CMD
3078  006344  066066                  DT13
3079  006346  066612                  DF21
3080                          ;ERR 67
3081  006350  060432                  EM20
3082  006352  063741                  DH17
3083  006354  066066                  DT13
3084  006356  066612                  DF21
3085                          ;ERR 70
3086  006360  062170                  EM74          ;RTZ NOT SET
3087  006362  065025                  DH41          ;DURING RECAL CMD
3088  006364  065530                  DT1
3089  006366  066462                  DF10
3090                          ;ERR 71
3091  006370  061402                  EM31          ;PORT NOT AVAIL
3092  006372  064502                  DH35          ;AFTER RLS
3093  006374  065530                  DT1
3094  006376  066462                  DF10
3095                          ;ERR 72
3096  006400  061361                  EM30          ;PIP SET
3097  006402  062714                  DH4           ;AFTER ATTN REC'D FROM RECAL
3098  006404  065530                  DT1
3099  006406  066462                  DF10
3100                          ;ERR 73
3101  006410  060115                  EM13          ;NO ATTN
3102  006412  063765                  DH18          ;AFTER UNLOAD CMD
3103  006414  065530                  DT1
3104  006416  066462                  DF10
3105                          ;ERR 74
3106  006420  060115                  EM13          ;NO ATTN
3107  006422  063310                  DH10          ;AT END OF HEAD LOADING
3108  006424  065530                  DT1
3109  006426  066462                  DF10
3110                          ;ERR 75
3111  006430  060475                  EM22          ;NO DRIVS IN $DEVM
3112  006432  062556                  DH1
3113  006434  065530                  DT1
3114  006436  066332                  DF1
3115                          ;ERR 76
3116  006440  060602                  EM23          ;NO DRIVS ON BUSS
3117  006442  062556                  DH1
3118  006444  065530                  DT1
3119  006446  066332                  DF1
3120                          ;ERR 77
3121  006450  060347                  EM17          ;AO ERROR
```

```
3122  006452  062760                    DH5                     ;AFTER TMO
3123  006454  066066                    DT13
3124  006456  066612                    DF21
3125                              ;ERR 100
3126  006460  060370                    EM18
3127  006462  062760                    DH5
3128  006464  066066                    DT13
3129  006466  066612                    DF21
3130                              ;ERR 101
3131  006470  060411                    EM19
3132  006472  062760                    DH5
3133  006474  066066                    DT13
3134  006476  066612                    DF21
3135                              ;ERR 102
3136  006500  060432                    EM20
3137  006502  062760                    DH5
3138  006504  066066                    DT13
3139  006506  066612                    DF21
3140                              ;ERR 103
3141  006510  061425                    EM32                    ;PORT AVAIL
3142  006512  063531                    DH14                    ;BEFORE TMO OR RLS
3143  006514  065530                    DT1
3144  006516  066462                    DF10
3145                              ;ERR 104
3146  006520  060347                    EM17                    ;AO ERROR
3147  006522  063563                    DH15                    ;WHILE PORT UNAVAIL
3148  006524  066066                    DT13
3149  006526  066612                    DF21
3150                              ;ERR 105
3151  006530  060370                    EM18
3152  006532  063563                    DH15
3153  006534  066066                    DT13
3154  006536  066612                    DF21
3155                              ;ERR 106
3156  006540  060411                    EM19
3157  006542  063563                    DH15
3158  006544  066066                    DT13
3159  006546  066612                    DF21
3160                              ;ERR 107
3161  006550  060432                    EM20
3162  006552  063563                    DH15
3163  006554  066066                    DT13
3164  006556  066612                    DF21
3165                              ;ERR 110
3166  006560  060115                    EM13                    ;NO ATTN
3167  006562  063612                    DH16                    ;TO ALLOW PORT TO SEIZE
3168  006564  065530                    DT1
3169  006566  066462                    DF10
3170                              ;ERR 111
3171  006570  061444                    EM33                    ;ATTN SET
3172  006572  064150                    DH23                    ;W/O REQUEST PENDING
3173  006574  065530                    DT1
3174  006576  066462                    DF10
3175                              ;ERR 112
3176  006600  060115                    EM13                    ;NO ATTN
3177  006602  062760                    DH5                     ;AFT TMO
```

```
3178  006604  065530                  DT1
3179  006606  066462                  DF10
3180                          ;ERR 113
3181  006610  061402                  EM31            ;PORT NO AVAIL
3182  006612  064116                  DH22            ;AFTER DRIVE CLEAR CMD
3183  006614  065530                  DT1
3184  006616  066462                  DF10
3185                          ;ERR 114
3186  006620  061467                  EM34            ;ATTN CLEARED
3187  006622  064200                  DH24            ;BY DR CLR TO OTHER PORT.
3188  006624  065530                  DT1
3189  006626  066462                  DF10
3190                          ;ERR 115
3191  006630  061444                  EM33            ;ATTN SET
3192  006632  064502                  DH35            ;AFT RELEASE ISSUED
3193  006634  065530                  DT1
3194  006636  066462                  DF10
3195                          ;ERROR 116
3196  006640  060057                  EM12            ;CONT NOT RDY
3197  006642  064012                  DH19            ;AFTER PACK CMD
3198  006644  065530                  DT1
3199  006646  066462                  DF10
3200                          ;ERROR 117
3201  006650  060057                  EM12            ;CONT NOT RDY
3202  006652  064035                  DH20            ;AFTER SEL DR CMD
3203  006654  065530                  DT1
3204  006656  066462                  DF10
3205                          ;ERROR 120
3206  006660  060057                  EM12
3207  006662  064070                  DH21            ;AFTER SUBSYS CLEAR
3208  006664  065530                  DT1
3209  006666  066462                  DF10
3210                          ;ERROR 121
3211  006670  060057                  EM12
3212  006672  063235                  DH9             ;AFTER START SPINDLE CMD
3213  006674  065530                  DT1
3214  006676  066462                  DF10
3215                          ;ERR 122
3216  006700  060115                  EM13            ;NO ATTN
3217  006702  064610                  DH37            ;AFT PLS & REQUEST PENDING
3218  006704  065530                  DT1
3219  006706  066462                  DF10
3220                          ;ERR 123
3221  006710  061467                  EM34            ;ATTN CLEARED
3222  006712  064502                  DH35            ;AFTER RELEASE
3223  006714  065530                  DT1
3224  006716  066462                  DF10
3225                          ;ERROR 124
3226  006720  060057                  EM12
3227  006722  063741                  DH17            ;AFTER RECAL CMD
3228  006724  065530                  DT1
3229  006726  066462                  DF10
3230                          ;ERR 125
3231  006730  062147                  EM73            ;CTO SET
3232  006732  062404                  EM84            ;WHILE WAITING FOR OR REC'D CONTR RDY.  MSG A&B BAD
3233  006734  065530                  DT1
```

```
3234  006736  066412                    DF5
3235                                ;ERR 126
3236  006740  062215                    EM79                    ;NED SET
3237  006742  062404                    EM84
3238  006744  065530                    DT1
3239  006746  066412                    DF5
3240                                ;ERR 127
3241  006750  057346                    EM5                     ;MDS SET
3242  006752  062404                    EM84
3243  006754  065530                    DT1
3244  006756  066412                    DF5
3245                                ;ERROR 130
3246  006760  061516                    EM35                    ;CERR NOT SET
3247  006762  064527                    DH36                    ;AFT SEL DRV CMD & NO DRA
3248  006764  065530                    DT1
3249  006766  066462                    DF10
3250                                ;ERROR 131
3251  006770  060057                    EM12                    ;NO RDY
3252  006772  064245                    DH25                    ;AFTER SEEK CMD
3253  006774  065530                    DT1
3254  006776  066462                    DF10
3255                                ;ERROR 132
3256  007000  060115                    EM13                    ;NO ATTN
3257  007002  064245                    DH25
3258  007004  065530                    DT1
3259  007006  066462                    DF10
3260                                ;ERR 133
3261  007010  060347                    EM17                    ;AO ERROR
3262  007012  064502                    DH35                    ;AFTER RLS ISSUED
3263  007014  066066                    DT13
3264  007016  066612                    DF21
3265                                ;ERR 134
3266  007020  060370                    EM18
3267  007022  064502                    DH35
3268  007024  066066                    DT13
3269  007026  066612                    DF21
3270                                ;ERR 135
3271  007030  060411                    EM19
3272  007032  064502                    DH35
3273  007034  066066                    DT13
3274  007036  066612                    DF21
3275                                ;ERR 136
3276  007040  060432                    EM20
3277  007042  064502                    DH35
3278  007044  066066                    DT13
3279  007046  066612                    DF21
3280                                ;ERROR 137
3281  007050  061713                    EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3282  007052  064245                    DH25
3283  007054  065530                    DT1
3284  007056  066462                    DF10
3285                                ;ERR 140
3286  007060  060347                    EM17                    ;MSG AO ERROR
3287  007062  065451                    DH51                    ;AFTER SEEK TO SELF
3288  007064  066066                    DT13
3289  007066  066612                    DF21
```

```
3290                                    ;ERR 141
3291  007070  060370                          EM18
3292  007072  065451                          DH51
3293  007074  066066                          DT13
3294  007076  066612                          DF21
3295                                    ;ERR 142
3296  007100  060411                          EM19
3297  007102  065451                          DH51
3298  007104  066066                          DT13
3299  007106  066612                          DF21
3300                                    ,ERR 143
3301  007110  060432                          EM20
3302  007112  065451                          DH51
3303  007114  066066                          DT13
3304  007116  066612                          DF21
3305                                    ;ERR 144
3306  007120  061607                          EM37                   ;MULT ATTN
3307  007122  062760                          DH5                    ;AFTER TMO
3308  007124  065530                          DT1
3309  007126  066462                          DF10
3310                                    ;ERR 145
3311  007130  060347                          EM17                   ;AO ERROR
3312  007132  064664                          DH38                   ;AFTER SEEK & IMMED RELEASE
3313  007134  066066                          DT13
3314  007136  066612                          DF21
3315                                    ;ERR 146
3316  007140  060370                          EM18
3317  007142  064664                          DH38
3318  007144  066066                          DT13
3319  007146  066612                          DF21
3320                                    ;ERR 147
3321  007150  060411                          EM19
3322  007152  064664                          DH38
3323  007154  066066                          DT13
3324  007156  066612                          DF21
3325                                    ;ERR 150
3326  007160  060432                          EM20
3327  007162  064664                          DH38
3328  007164  066066                          DT13
3329  007166  066612                          DF21
3330                                    ;ERROR 151
3331  007170  060057                          EM12                   ;NO RDY
3332  007172  064116                          DH22                   ;AFTER CLEAR CMD
3333  007174  065530                          DT1
3334  007176  066462                          DF10
3335                                    ;ERR 152
3336  007200  060115                          EM13                   ;NO ATTN
3337  007202  064664                          DH38                   ;AFTER SEEK & IMMED RLS
3338  007204  065530                          DT1
3339  007206  066462                          DF10
3340                                    ;ERR 153
3341  007210  061640                          EM38                   ;RE-TRIGGER NO GOOD
3342  007212  065172                          DH43                   ;AFTER RE-SEIZE MIDWAY THRU TMO
3343  007214  065530                          DT1
3344  007216  066462                          DF10
3345                                    ;ERROR 154
```

```
 3346  007220  062020                          EM55                    ;ATTN NOT CLEARED
 3347  007222  064116                          DH22
 3348  007224  065530                          DT1
 3349  007226  066462                          DF10
 3350                                ;ERR 155
```

```
3351  007230  061313                  EM29            ;PORT AVAIL-TIMERS NOT INHIB.
3352  007232  065330                  DH45            ;BEFORE RLS WHEN UNLOADED
3353  007234  065530                  DT1
```

```
3354  007236  066462                    DF10
3355                              ;ERR 156
3356  007240  060115                    EM13                    ;NO ATTN
3357  007242  064245                    DH25                    ;AFTER SEEK CMD
3358  007244  065530                    DT1
3359  007246  066462                    DF10
3360                              ;ERR 157
3361  007250  061402                    EM31                    ;PORT NOT AVAIL
3362  007252  064245                    DH25                    ;AFTER SEEK CMD
3363  007254  065530                    DT1
3364  007256  066462                    DF10
3365                              ;ERROR 160
3366  007260  061444                    EM33                    ;ATTN SET
3367  007262  064664                    DH38                    ;AFTER SEEK & IMMED RLS
3368  007264  065530                    DT1
3369  007266  066462                    DF10
3370                              ;ERR 161
3371  007270  060347                    EM17                    ;MSG A0 ERROR
3372  007272  064245                    DH25                    ;AFTER SEEK CMD
3373  007274  066066                    DT13
3374  007276  066612                    DF21
3375                              ;ERR 162
3376  007300  060370                    EM18
3377  007302  064245                    DH25
3378  007304  066066                    DT13
3379  007306  066612                    DF21
3380                              ;ERR 163
3381  007310  060411                    EM19
3382  007312  064245                    DH25
3383  007314  066066                    DT13
3384  007316  066612                    DF21
3385                              ;ERR 164
3386  007320  060432                    EM20
3387  007322  064245                    DH25
3388  007324  066066                    DT13
3389  007326  066612                    DF21
3390                              ;ERR 165
3391  007330  060347                    EM17                    ;MSG A0 ERROR
3392  007332  062760                    DH5                     ;AFTER TM0
3393  007334  066066                    DT13
3394  007336  066612                    DF21
3395                              ;ERR 166
3396  007340  060370                    EM18
3397  007342  062760                    DH5
3398  007344  066066                    DT13
3399  007346  066612                    DF21
3400                              ;ERR 167
3401  007350  060411                    EM19
3402  007352  062760                    DH5
3403  007354  066066                    DT13
3404  007356  066612                    DF21
3405                              ;ERR 170
3406  007360  060432                    EM20
3407  007362  062760                    DH5
3408  007364  066066                    DT13
3409  007366  066612                    DF21
```

```
3410                                    ;ERROR 171
3411  007370  060057                           EM12                    ;NO RDY
3412  007372  064371                           DH30                    ;AFTER READ HEADER CMD
3413  007374  065530                           DT1
3414  007376  066462                           DF10
3415                                    ;ERROR 172
3416  007400  061402                           EM31                    ;PORT NOT AVAIL
3417  007402  064502                           DH35                    ;AFT RELEASE
3418  007404  065530                           DT1
3419  007406  066462                           DF10
3420                                    ,ERROR 173
3421  007410  062053                           EM63                    ;DLT SET
3422  007412  064371                           DH30
3423  007414  065530                           DT1
3424  007416  066522                           DF15
3425                                    ;ERROR 174
3426  007420  060453                           EM21                    ;CERR SET
3427  007422  064371                           DH30
3428  007424  065530                           DT1
3429  007426  066522                           DF15
3430                                    ;ERROR 175
3431  007430  061713                           EM39                    ;CYL DIFF NOT CLEARED
3432  007432  063310                           DH10                    ;AT END OF HEAD LOADING
3433  007434  065530                           DT1
3434  007436  066462                           DF10
3435                                    ;ERROR 176
3436  007440  061762                           EM40                    ;CYL ADDR NOT CLEARED.
3437  007442  063310                           DH10
3438  007444  065530                           DT1
3439  007446  066462                           DF10
3440                                    ;ERROR 177
3441  007450  061313                           EM29                    ;PORT AVAIL-TIMERS NOT INHIB
3442  007452  065025                           DH41                    ;DURING RECAL CMD
3443  007454  065530                           DT1
3444  007456  066462                           DF10
3445                                    ;ERROR 200
3446  007460  060057                           EM12                    ;NO RDY
3447  007462  064736                           DH39                    ;AFTER WRITE HEADER CMD
3448  007464  065530                           DT1
3449  007466  066522                           DF15
3450                                    ;ERROR 201
3451  007470  060453                           EM21                    ;CERR SET
3452  007472  064736                           DH39
3453  007474  065530                           DT1
3454  007476  066522                           DF15
3455                                    ;ERROR 202
3456  007500  062074                           EM65                    ;READ HEADER ERROR
3457  007502  062556                           DH1
3458  007504  065706                           DT7
3459  007506  066502                           DF14
3460                                    ;ERROR 203
3461  007510  061402                           EM31                    ;PORT NOT AVAIL
3462  007512  065025                           DH41                    ;DURING RECAL CMD
3463  007514  065530                           DT1
3464  007516  066462                           DF10
3465                                    ;ERROR 204
```

```
3466  007520  000000                    0
3467  007522  000000                    0
3468  007524  000000                    0
3469  007526  000000                    0
3470                              ;ERROR 205
3471  007530  000000                    0
3472  007532  000000                    0
3473  007534  000000                    0
3474  007536  000000                    0
3475                              ;ERROR 206
3476  007540  061544                    EM36          ;CYL ADDR IN RKMR3 INCORRECT
3477  007542  064245                    DH25          ;AFTER SEEK CMD
3478  007544  065754                    DT8
3479  007546  066436                    DF6
3480                              ;ERROR 207
3481  007550  061544                    EM36          ;CYL ADDR IN RKMR3 INCORRECT
3482  007552  064245                    DH25          ;AFTER SEEK CMD
3483  007554  065574                    DT4
3484  007556  066436                    DF6
3485                              ;ERROR 210
3486  007560  060453                    EM21          ;CERR SET
3487  007562  064245                    DH25
3488  007564  065530                    DT1
3489  007566  066462                    DF10
3490                              ;ERR 211
3491  007570  060347                    EM17          ;MSG AO ERROR
3492  007572  064502                    DH35          ;AFTER RLS ISSUED
3493  007574  066066                    DT13
3494  007576  066612                    DF21
3495                              ;ERR 212
3496  007600  060370                    EM18
3497  007602  064502                    DH35
3498  007604  066066                    DT13
3499  007606  066612                    DF21
3500                              ;ERR 213
3501  007610  060411                    EM19
3502  007612  064502                    DH35
3503  007614  066066                    DT13
3504  007616  066612                    DF21
3505                              ;ERR 214
3506  007620  060432                    EM20
3507  007622  064502                    DH35
3508  007624  066066                    DT13
3509  007626  066612                    DF21
3510                              ;ERROR 215
3511  007630  000000                    0
3512  007632  000000                    0
3513  007634  000000                    0
3514  007636  000000                    0
3515                              ;ERROR 216
3516  007640  000000                    0
3517  007642  000000                    0
3518  007644  000000                    0
3519  007646  000000                    0
3520                              ;ERROR 217
3521  007650  000000                    0
```

```
3522  007652  000000                    0
3523  007654  000000                    0
3524  007656  000000                    0
3525                              ;ERROR 220
3526  007660  000000                    0
3527  007662  000000                    0
3528  007664  000000                    0
3529  007666  000000                    0
3530                              ;ERROR 221
3531  007670  060347                    EM17                ;MSG A0 ERROR
3532  007672  063741                    DH17
3533  007674  066066                    DT13
3534  007676  066612                    DF21
3535                              ;ERROR 222
3536  007700  060411                    EM19                ;MSG A1 ERROR
3537  007702  063741                    DH17
3538  007704  066066                    DT13
3539  007706  066612                    DF21
3540                              ;ERROR 223
3541  007710  060453                    EM21                ;CERR SET
3542  007712  063042                    DH7                 ;AFT SCLR
3543  007714  065530                    DT1
3544  007716  066462                    DF10
3545                              ;ERR 224
3546  007720  061544                    EM36                ;CYL ADDR IN RKMR3 BAD
3547  007722  063354                    DH12                ;AFT SEEK CMD
3548  007724  065574                    DT4
3549  007726  066436                    DF6
3550                              ;ERROR 225
3551  007730  000000                    0
3552  007732  000000                    0
3553  007734  000000                    0
3554  007736  000000                    0
3555                              ;ERROR 226
3556  007740  060057                    EM12                ;NO RDY
3557  007742  064270                    DH26                ;AFTER READ DATA CMD
3558  007744  065530                    DT1
3559  007746  066462                    DF10
3560                              ;ERROR 227
3561  007750  060453                    EM21                ;CERR SET
3562  007752  064270                    DH26
3563  007754  065530                    DT1
3564  007756  066522                    DF15
3565                              ;ERROR 230
3566  007760  060304                    EM16                ;CANNOT READ BSE INFO
3567  007762  063445                    DH13                ;ON SEC 10. 12. 14. 16. 18. 20
3568  007764  065530                    DT1
3569  007766  066542                    DF17
3570                              ;ERROR 231
3571  007770  000000                    0
3572  007772  000000                    0
3573  007774  000000                    0
3574  007776  000000                    0
3575                              ;ERROR 232
3576  010000  000000                    0
3577  010002  000000                    0
```

```
3578  010004  000000                          0
3579  010006  000000                          0
3580                                ;ERROR 233
3581  010010  060304                          EM16                    ;CANNOT READ BSE INFO
3582  010012  065121                          DH42                    ;ON SECT 0,2,4,6,8
3583  010014  065530                          DT1
3584  010016  066542                          DF17
3585                                ;ERROR 234
3586  010020  000000                          0
3587  010022  000000                          0
3588  010024  000000                          0
3589  010026  000000                          0
3590                                ;ERROR 235
3591  010030  062116                          EM69                    ;ALIGN CARTRIDGE USED
3592  010032  065251                          DH44                    ;WILL BYPASS FORMAT & ALL R/W TESTS
3593  010034  065530                          DT1
3594  010036  066462                          DF10
3595
3596                                .SBTTL  PROGRAM SETUP
3597
3598  010040  012737  000001  001336  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START:  INPUT PARAMETERS
3599  010046  000402                          BR      PRGSRT          ;START PROGRAM
3600
3601  010050  005037  001336          START:  CLR     PARAM           ;200 START, DEFAULT
3602  010054  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
3603  010056  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3604  010062  012746  000000                  MOV     #PR0,-(SP)      ;PSW LOADED TO BE
3605  010066  012746  010074                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
3606  010072  000002                          RTI                     ;ENABLE ALL INTERRUPTS
3607
3608  010074  004737  052260          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
3609                                                                   ;& TURN ON KB INTERRUPT
3610
3611
3612                                ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3613                                ;*** ANY DEVICE WHICH SETS ITS   ***
3614                                ;*** INTERRUPT ENABLE BIT WILL   ***
3615                                ;*** SERVICED.                   ***
3616
3617                                ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3618                                ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3619                                ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3620
3621                                ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3622
3623                                                                   ;SYSMAC 'SETUP'
3624                                .SBTTL  INITIALIZE THE COMMON TAGS
3625                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3626  010100  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3627  010104  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
3628  010106  022706  001140                  CMP     #SWR,R6 ;;DONE?
3629  010112  001374                          BNE     .-6             ;;LOOP BACK IF NO
3630  010114  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3631                                ;;INITIALIZE A FEW VECTORS
3632  010120  012737  050366  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3633  010126  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
```

```
3634  010134  012737  050646  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3635  010142  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
3636  010150  012737  054476  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3637  010156  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
3638  010164  012737  050122  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
3639  010172  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
3640  010200  013737  042740  042732          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
3641  010206  005037  001174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
3642  010212  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3643  010216  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
3644  010224  012737  010224  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3645  010232  012737  010232  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
3646                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER.  IF NOT FOUND OR IT IS
3647                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3648  010240  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3649  010244  012737  010300  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3650  010252  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
3651  010260  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3652  010266  022777  177777  170644          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3653  010274  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3654                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
3655  010276  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
3656  010300  012716  010306          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
3657  010304  000002                          RTI
3658  010306  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
3659  010314  012737  000174  001142          MOV     #DISPREG,DISPLAY
3660  010322  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
3661
3662  010326  005037  001216                  CLR     $PASS           ;;CLEAR PASS COUNT
3663  010332  132737  000200  001231          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
3664  010340  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
3665  010342  012737  001232  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
3666  010350                          67$:
3667  010350  012737  010414  000004  MEMPAR: MOV     #1$,ERRVEC      ;SETUP TIMEOUT VECTOR
3668  010356  012737  000340  000006          MOV     #PR7,ERRVEC+2
3669
3670  010364  012701  172100                  MOV     #MEMBAS,R1      ;ADDR OF MEM CSR
3671  010370  005011                  3$:     CLR     (R1)            ;SEE IF CAN REFERENCE
3672  010372  012711  000001                  MOV     #1,(R1)         ;SET ENABLE BIT IF YES
3673  010376  012737  050024  000114          MOV     #MEMERR,MEMVEC  ;LOAD VECTOR IF NO TIMEOUT
3674  010404  012737  000340  000116          MOV     #PR7,MEMVEC+2
3675  010412  000401                          BR      2$
3676
3677  010414  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
3678  010416  062701  000002          2$:     ADD     #2,R1           ;TRY NEXT CSR
3679  010422  020127  172140                  CMP     R1,#MEMBAS+40   ;SEE IF TRIED ALL
3680  010426  001360                          BNE     3$              ;BR IF NO
3681  010430  012737  000006  000004          MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
3682  010436  005037  000006                  CLR     ERRVEC+2
3683
3684  010442  004737  043026                  JSR     PC,CLRFLG       ;CLEAR DDUMP THRU UNITB
3685  010446  005037  001220                  CLR     $DEVCT
3686  010452  005037  001222                  CLR     $UNIT
3687
3688                                  ;
3689                                  ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
```

```
                                        ;
                                        ;
010456  005737  000042          START1: TST     42
010462  001014                          BNE     1$              ;BR IF AUTO
010464  004737  043046                  JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
010470  123727  000041  000013          CMPB    41,#13          ;13=LOADED BY XXDP
010476  001010                          BNE     2$
010500  005237  005424                  INC     DDUMP           ;SET RK06 DUMP MODE FLAG
010504  104401  055743                  TYPE    ,MSG2           ;REPLACE DRO PACK W/SCRATCH & DO<CR>
010510  000137  010524                  JMP     ST2
010514  000137  010570          1$:     JMP     ST3
010520  005237  005432          2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG
                                        ;
                                        ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
                                        ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
                                        ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
                                        ;       EX:     DRIVES TO BE TESTED:  0,2,4<CR>
                                        ;FOR DUAL PORT, ONLY EVEN NUMBERED DRIVES MUST BE TESTED
                                        ;
010524  005737  001336          ST2:    TST     PARAM
010530  001002                          BNE     1$              ;BR IF 220 START
010532  000137  010622                  JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
010536  104401  056014          1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
010542  004737  043126                  JSR     PC,GDRVS        ;GET DR NOS.
010546  104401  056067                  TYPE    ,MSG4           ;BUSS ADDR
010552  004737  043274                  JSR     PC,GBA          ;GET BA
010556  104401  056121                  TYPE    ,MSG5           ;CONT INT VECTOR
010562  004737  043322                  JSR     PC,GINT         ;GET INT VECTOR
010566  000427                          BR      ST5
                                        ;
                                        ;AUTO MODE
                                        ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
                                        ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
                                        ;ON THE BUSS
                                        ;
010570  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
010576  001007                          BNE     1$
010600  005237  005426                  INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
010604  004737  043046                  JSR     PC,TITLE
010610  104401  056223                  TYPE    ,MSG7           ;DRO NOT TSTD
010614  000402                          BR      ST4
010616  005237  005430          1$:     INC     ACT11           ;SET ACT AUTO FLAG.
                                        ;
010622  012737  177440  001264  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
010630  012737  000210  001314          MOV     #210,RKVEC      ;DEFAULT VALUE
010636  004737  043354                  JSR     PC,SETINT
010642  005237  005462                  INC     SIZFLG          ;DO "SIZE THE BUSS" TEST
                                        ;
010646  005037  005276          ST5:    CLR     UNLD            ;INITIALIZE FLAGS
010652  005037  005300                  CLR     BADHDR          ;USED IN 'STOP' ROUTINE
010656  005037  005302                  CLR     HPEND           ;FOR VALID PROGRAM HALTS
010662  005037  001176                  CLR     $ESCAPE
010666  005037  001172                  CLR     $TMP5           ;CLEAR RK07 FLAG
```

```
3746  010672  012737  005436  001342          MOV     #DRIVD,DRVPTR   ;SETUP
3747  010700  005037  001220                  CLR     $DEVCT          ;NO. OF DRVS DONE
3748  010704  005037  001222                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
3749  010710  005037  005464                  CLR     UNITB           ;PORT A TESTING
3750  010714  112737  000101  056576          MOVB    #'A,MSG19A      ;PORT A TESTING
3751  010722  012737  010770  000004          MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
3752  010730  005777  170372                  TST     @LKS            ;SEE IF L-CLOCK THERE
3753  010734  005237  005456                  INC     LCLKF           ;PRESENT, SET FLAG.
3754  010740  013700  001330                  MOV     LCVEC,R0        ;VECTOR ADDR
3755  010744  012737  011026  000004          MOV     #2$,ERRVEC
3756  010752  005777  170342                  TST     @PKS            ;SEE IF P-CLOCK THERE
3757  010756  005237  005460                  INC     PCLKF           ;PRESENT, SET FLAG
3758  010762  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3759  010766  000412                          BR      3$
3760
3761  010770  022626                  1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
3762  010772  012737  011032  000004          MOV     #4$,ERRVEC
3763  011000  005777  170314                  TST     @PKS            ;SEE IF P-CLOCK THERE
3764  011004  005237  005460                  INC     PCLKF           ;PRESENT, SET FLAG
3765  011010  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
3766  011014  012720  047154          3$:     MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
3767  011020  012710  000300                  MOV     #PR6,(R0)
3768  011024  000407                          BR      TST1            ;;GO TO NEXT TEST
3769
3770  011026  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
3771  011030  000771                          BR      3$
3772
3773  011032  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
3774  011034  104401  056376                  TYPE    .MSG13          ;NO CLOCKS PRESENT TESTS BYPASSED
3775  011040  000137  042644                  JMP     $EOP
3776
3777
```

```
                                .SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

                                ;;**********************************************************************
                                ;*TEST 1        REFERENCE ALL CONTROLLER REGISTERS
                                ;*
                                ;*      THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
                                ;*      CAN BE ACCESSED.  THE INABILITY TO BE ACCESSED WILL
                                ;*      RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE.  ANY
                                ;*      ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                                ;*      TESTS AND JUMPING TO 'END OF PASS'
                                ;*
                                ;;**********************************************************************
011044  000004                TST1:   SCOPE
011046  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
011054  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

011060  012746  000000                MOV     #PR0,-(SP)      ;RESET PSW TO PRIORITY 0
011064  012746  011072                MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
011070  000002                        RTI
011072                          5$:

011072  012737  011216  000004        MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
011100  013705  001264                MOV     $BASE,R5        ;SETUP INDEX REG.
011104  005765  000000                TST     RKCS1(R5)       ;REFERENCE ALL THE
011110  005765  000010                TST     RKCS2(R5)       ;CONTROLLER REGISTERS
011114  005765  000002                TST     RKWC(R5)
011120  005765  000004                TST     RKBA(R5)
011124  005765  000006                TST     RKDA(R5)
011130  005765  000012                TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
011134  005765  000014                TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
011140  005765  000016                TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
011144  005765  000020                TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
011150  005765  000024                TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
011154  005765  000026                TST     RKMR1(R5)
011160  005765  000034                TST     RKMR2(R5)
011164  005765  000036                TST     RKMR3(R5)
011170  005765  000030                TST     RKECPS(R5)
011174  005765  000032                TST     RKECPT(R5)

011200  012737  047736  000004        MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
011206  012737  000340  000006        MOV     #PR7,ERRVEC+2
011214  000404                        BR      TST2            ;;GO TO NEXT TEST

011216  022626                  1$:   CMP     (SP)+,'SP)+     ;RESTORE STACK POINTER
011220  104007                        ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
011222  000137  042704                JMP     $EOP1


                                ;;**********************************************************************
                                ;*TEST 2        SIZE THE BUSS
                                ;*
                                ;*      THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                                ;*      EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                                ;*      MANUAL MODE.
                                ;*      EVERY EVEN NUMBERED DRIVE (0,2,4,6) IS ADDRESSED.
                                ;*      CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
```

```
3834                                    ;*      DRIVE WILL BE TESTED AS AN RK06.  IF SET, THE PROGRAM WILL BYPASS
3835                                    ;*      TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
3836                                    ;*      MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
3837                                    ;*      DICATING THE OTHER PORT IS ACCESSED.
3838                                    ;*      IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS AN RK07.
3839                                    ;*
3840                                    ;;**********************************************************************
3841  011226  000004                    TST2:   SCOPE
3842  011230  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
3843  011236  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
3844
3845  011242  005237  001456                    INC     BYPCERR         ;DO NOT DO 'CKCERR' ROUTINE
3846
3847
3848  011246  132737  000200  001231            BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
3849  011254  001002                            BNE     14$             ;BR IF YES
3850  011256  000137  011420                    JMP     12$             ;ELSE DO NORM SIZING OR VERIFY
3851
3852  011262  104401  056311            14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
3853  011266  005037  005434                    CLR     DRIVS           ;# OF DRIVES PRESENT
3854  011272  005000                            CLR     R0              ;DRV ADDR
3855  011274  012701  005436                    MOV     #DRIV0,R1       ;DRV FLAG
3856  011300  013702  001266                    MOV     $DEVM,R2        ;APT DEVICE MAP
3857
3858  011304  032702  000001            15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
3859  011310  001420                            BEQ     16$             ;BR IF NO
3860  011312  022700  000001                    CMP     #BIT0,R0        ;SEE IF ODD # DRIVE
3861  011316  001405                            BEQ     11$             ;BR IF NO
3862
3863  011320  104401  061205                    TYPE    ,EM28           ;ONLY EVEN # ALLOWED IN $DEVM
3864  011324  000000                            HALT                    ;RELOAD $DEVM & PRESS 'CONTINUE'
3865  011326  000137  010054                    JMP     PRGSRT          ;RESTART
3866  011332  005237  005434            11$:    INC     DRIVS           ;ELSE INCR DRIVE COUNT
3867  011336  005211                            INC     (R1)            ;& SET DRIVE PRESENT FLAG
3868  011340  104401  001205                    TYPE    ,$CRLF
3869  011344  010046                            MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3870                                            ;;TYPE DRIVE #
3871  011346  104403                            TYPOS                   ;;GO TYPE--OCTAL ASCII
3872  011350     001                            .BYTE   1               ;;TYPE 1 DIGIT(S)
3873  011351     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
3874
3875  011352  022121            16$:    CMP     (R1)+,(R1)+     ;ADV PTR TO NEXT EVEN #
3876  011354  062700  000002                    ADD     #2,R0           ;ADD DRV ADDR TO NEXT EVEN #
3877  011360  022700  000010                    CMP     #8.,R0          ;ALL 4 TESTED?
3878  011364  001402                            BEQ     17$             ;BR IF YES
3879
3880  011366  006002                            ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
3881  011370  000745                            BR      15$             ;& TRY AGAIN
3882
3883  011372  005737  005434            17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
3884  011376  001402                            BEQ     18$             ;BR IF NO
3885  011400  000137  012076                    JMP     VERIFY          ;ELSE EXIT TEST & SETUP FOR RK07'S
3886
3887  011404  104075            18$:    ERROR   75              ;NO DRIVES FOUND IN $DEVM
3888  011406  000000                            HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
3889  011410  000137  010646                    JMP     ST5             ;TO TRY AGAIN
```

```
3890  011414  000137  012076          20$:    JMP     VERIFY          ;DO NOT SIZE, GO TO NEXT TEST
3891
3892  011420  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3893  011426  013737  001400  005352          MOV     T10,TEMP1       ;SET TIMEOUT
3894  011434  004737  043466                  JSR     PC,FRDY         ;FIND RDY
3895  011440  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
3896  011442  005737  005462                  TST     SIZFLG          ;SIZE BUS?
3897  011446  001762                          BEQ     20$             ;BR IF NO
3898  011450  104401  056311                  TYPE    ,MSG10          ;WILL TEST DRIVES
3899  011454  005037  005434                  CLR     DRIVS           ;# OF DRIVES PRESENT
3900  011460  005000                          CLR     R0              ;DRV ADDR
3901  011462  012701  005436                  MOV     #DRIVO,R1       ;DRV FLAG
3902  011466                          1$:
3903  011466  104415                          SCOP1
3904  011470  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3905
3906  011474  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3907  011502  013737  001400  005352          MOV     T10,TEMP1       ;SET TIMEOUT
3908  011510  004737  043466                  JSR     PC,FRDY         ;FIND RDY
3909  011514  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
3910  011516  010065  000010                  MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
3911  011522  012737  000001  005314          MOV     #SELDRV,HCS1
3912  011530  053737  001170  005314          BIS     $TMP4,HCS1      ;ADD CDT IF RK07
3913  011536  013765  005314  000000          MOV     HCS1,RKCS1(R5)  ;GET STATUS
3914  011544  013737  001412  005352          MOV     T50000,TEMP1
3915  011552  004737  044242                  JSR     PC,DLY          ;DO DELAY TO CATCH MDS
3916  011556  013737  001400  005352          MOV     T10,TEMP1
3917  011564  004737  043466                  JSR     PC,FRDY         ;FIND RDY
3918  011570  104117                          ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
3919  011572  032737  100000  005314          BIT     #CERR,HCS1
3920  011600  001053                          BNE     2$
3921  011602  013737  005342  005352          MOV     HMR2,TEMP1
3922  011610  042737  177770  005352          BIC     # C<DRVMSK>,TEMP1
3923  011616  020037  005352                  CMP     R0,TEMP1        ;S/B SAME
3924  011622  001020                          BNE     3$
3925  011624  005700                          TST     R0
3926  011626  001003                          BNE     4$
3927  011630  005737  005426                  TST     DDPCH           ;SEE IF XXDP CHAIN MODE
3928  011634  001016                          BNE     5$
3929  011636  005237  005434          4$:     INC     DRIVS           ;INC DRIVE COUNT.
3930  011642  005211                          INC     (R1)            ;SET DRIVE PRESENT FLAG
3931  011644  053711  001170                  BIS     $TMP4,(R1)      ;ADD CDT IF RK07.
3932  011650  104401  001205                  TYPE    ,$CRLF
3933  011654  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3934                                                                  ;;TYPE DR #
3935  011656  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3936  011660     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3937  011661     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3938  011662  000403                          BR      5$
3939
3940  011664  004737  044260          3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
3941  011670  104001                          ERROR   1               ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
3942
3943  011672  022121                  5$:     CMP     (R1)+,(R1)+     ;ADV PTR TO NEXT EVEN #
3944  011674  062700  000002                  ADD     #2,R0           ;ADV DRV ADDR TO NEXT EVEN #
3945  011700  005037  001170                  CLR     $TMP4           ;CLEAR FOR NEXT TRY
```

```
3946  011704  022700  000010              CMP     #8.,R0          ;TESTED ALL 4 DRIVES?
3947  011710  001266                      BNE     1$              ;BR IF NO
3948  011712  005737  005434              TST     DRIVS
3949  011716  001065                      BNE     10$
3950  011720  104076                      ERROR   76              ;NO DRIVES FOUND ON BUSS
3951  011722  000000                      HALT                    ;SETUP CORRECTLY
3952  011724  000137  010646              JMP     ST5             ;AND PRESS 'CONTINUE'
3953
3954  011730  032737  000040  005330  2$: BIT     #DTYE,HER
3955  011736  001405                      BEQ     13$
3956  011740  012737  002000  001170      MOV     #CDT,$TMP4      ;ADD CDT
3957  011746  000137  011466              JMP     1$              ;TRY AGAIN
3958  011752  032737  001000  005316  13$: BIT    #MDS,HCS2
3959  011760  001015                      BNE     6$
3960  011762  032737  000400  005316      BIT     #UFE,HCS2
3961  011770  001015                      BNE     7$
3962  011772  032737  000001  005326      BIT     #DRA,HDS
3963  012000  001015                      BNE     8$
3964  012002  032737  010000  005316      BIT     #NED,HCS2
3965  012010  001424                      BEQ     9$
3966  012012  000727                      BR      5$
3967
3968  012014  004737  044260          6$: JSR     PC,BYP          ;TYPE BYP DR #
3969  012020  104002                      ERROR   2               ;MDS DETECTED
3970  012022  000723                      BR      5$
3971
3972  012024  004737  044260          7$: JSR     PC,BYP
3973  012030  104003                      ERROR   3               ;UFE DETECTED
3974  012032  000717                      BR      5$
3975
3976  012034  032737  010000  005316  8$: BIT     #NED,HCS2
3977  012042  001675                      BEQ     4$
3978  012044  104401  056500              TYPE    ,MSG15          ;DRV#
3979  012050  010046                      MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3980                                                              ;;TYPE DR#
3981  012052  104403                      TYPOS                   ;;GO TYPE--OCTAL ASC!!
3982  012054     001                      .BYTE   1               ;;TYPE 1 DIGIT(S)
3983  012055     000                      .BYTE   0               ;;SUPPRESS LEADING ZEROS
3984  012056  104010                      ERROR   10              ;DRA & NED BOTH SET
3985  012060  000704                      BR      5$
3986
3987  012062  004737  044260          9$: JSR     PC,BYP
3988  012066  104004                      ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
3989  012070  000700                      BR      5$
3990  012072  000137  012472          10$ JMP     NUDRV
3991
3992  012076                          VERIFY:
3993
3994
3995                                    ;;*****************************************************************
3996                                    ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
3997                                    ;*
3998                                    ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
3999                                    ;*      DEFAULTED.  EVERY EVEN NUMBERED DRIVE (0,2,4,6) IS ADDRESSED &
4000                                    ;*      CONTROLLER ERROR (CERR) IS EXAMINED.  IF NOT SET, THE
4001                                    ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06
```

```
                                        ;*      IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
                                        ;*     ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                                        ;*     NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                                        ;*     NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFORMATION TO
                                        ;*     VERIFY IT WAS NOT SPECIFIED.
                                        ;*     IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS AN RK07.
                                        ;*
                                        ;;**********************************************************************
012076  000004                  TST3:   SCOPE
012100  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
012106  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
012112  005000                          CLR     R0              ;DRIVE ADDR
012114  012701  005436                  MOV     #DRIV0,R1       ;DRIVE FLAG
012120                          1$:
012120  104415                          SCOP1
012122  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

012126  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
012134  013737  001400  005352          MOV     T10,TEMP1       ;SET TIME OUT
012142  004737  043466                  JSR     PC,FRDY         ;FIND RDY
012146  104120                          ERROR   120             ;NO RDY AFTER SCLR
012150  010065  000010                  MOV     R0,RKCS2(R5)    ;DRV ADDR
012154  012737  000001  005314          MOV     #SELDRV,HCS1
012162  053737  001170  005314          BIS     $TMP4,HCS1      ;ADD CDT IF RK07
012170  013765  005314  000000          MOV     HCS1,RKCS1(R5)  ;GET STATUS
012176  013737  001412  005352          MOV     T50000,TEMP1
012204  004737  044242                  JSR     PC,DLY          ;DO DELAY TO CATCH MDS
012210  013737  001400  005352          MOV     T10,TEMP1
012216  004737  043466                  JSR     PC,FRDY         ;FIND RDY
012222  104117                          ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
012224  032737  100000  005314          BIT     #CERR,HCS1
012232  001037                          BNE     2$
012234  013737  005342  005352          MOV     HMR2,TEMP1
012242  042737  177770  005352          BIC     # C<DRVMSK>,TEMP1
012250  020037  005352                  CMP     R0,TEMP1        ;S/B SAME
012254  001015                          BNE     3$
012256  005711                  11$:    TST     (R1)
012260  001402                          BEQ     4$
012262  053711  001170                  BIS     $TMP4,(R1)      ;SET RK07 FLAG
012266  022121                  4$:     CMP     (R1)+,(R1)+     ;ADV PTR TO NEXT EVEN #
012270  062700  000002                  ADD     #2,R0           ;ADV DRV ADDR TO NEXT EVEN #
012274  005037  001170                  CLR     $TMP4           ;CLEAR FOR NEXT DRIVE
012300  022700  000010                  CMP     #8.,R0
012304  001305                          BNE     1$              ;MORE LEFT
012306  000475                          BR      TST4            ;;GO TO NEXT TEST

012310  004737  044260          3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
012314  104001                          ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
012316  005711                          TST     (R1)
012320  001762                          BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
012322  005337  005434          12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
012326  005011                          CLR     (R1)            ;CLEAR DRIVE FLAG
012330  000756                          BR      4$

012332  032737  000040  005330  2$:     BIT     #DTYE,HER
012340  001405                          BEQ     13$
```

```
4058  012342  012737  002000  001170          MOV     #CDT,$TMP4      ;ADD CDT
4059  012350  000137  012120                  JMP     1$              ;TRY AGAIN
4060
4061  012354  032737  001000  005316  13$:    BIT     #MDS,HCS2
4062  012362  001027                          BNE     6$
4063  012364  032737  000400  005316          BIT     #UFE,HCS2
4064  012372  001027                          BNE     7$
4065  012374  032737  000001  005326          BIT     #DRA,HDS
4066  012402  001005                          BNE     8$
4067  012404  032737  010000  005316          BIT     #NED,HCS2
4068  012412  001423                          BEQ     9$
4069  012414  000404                          BR      10$
4070  012416  032737  010000  005316  8$:     BIT     #NED,HCS2
4071  012424  001714                          BEQ     11$
4072  012426  005711                  10$:    TST     (R1)
4073  012430  001716                          BEQ     4$
4074
4075  012432  004737  044260                  JSR     PC,BYP          ;TYPE BYPASS DRIVE#
4076  012436  104006                          ERROR   6
4077  012440  000730                          BR      12$
4078
4079  012442  004737  044260          6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
4080  012446  104002                          ERROR   2               ;MDS DETECTED
4081  012450  000724                          BR      12$
4082
4083  012452  004737  044260          7$:     JSR     PC,BYP
4084  012456  104003                          ERROR   3               ;UFE DETECTED
4085  012460  000720                          BR      12$
4086
4087  012462  004737  044260          9$:     JSR     PC,BYP
4088  012466  104004                          ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
4089  012470  000714                          BR      12$
4090
4091
4092
4093                                          ;
4094                                          ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
4095                                          ;DRIVE PRESENT
4096                                          ;
4097                                          ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
4098                                          ;UNDER TEST
4099
4100  012472  005037  001456          NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
4101                                                                  ;ALLOW CHECKING CERR IN 'FRDY'
4102  012476  005037  001170                  CLR     $TMP4           ;CLEAR RK07 FLAG
4103
4104                                  ;;**************************************************************************
4105                                  ;*TEST 4         FIND NEXT DRIVE TO BE TESTED
4106                                  ;*
4107                                  ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
4108                                  ;*      ADDRESS IN '$UNIT' & $TMP4 IS SET TO CDT IF RK07.
4109                                  ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
4110                                  ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
4111                                  ;*
4112                                  ;;**************************************************************************
4113  012502  000004                  TST4:   SCOPE
```

FIND NEXT DRIVE TO BE TESTED

```
4114  012504  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4115  012512  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4116  012516  012737  000004  001214          MOV     #$TN-1,$TESTN
4117  012524  012737  000004  001102          MOV     #$TN-1,$TSTNM
4118
4119  012532  005737  005434                  TST     DRIVS           ;ANY DRIVES PRESENT?
4120  012536  001004                          BNE     4$              ;YES BRANCH
4121  012540  104401  056260                  TYPE    ,MSG8           ;ALL DRIVES TESTED
4122  012544  000137  042704                  JMP     $EOP1           ;NO, GO TO END
4123
4124  012550  005037  005464          4$:     CLR     UNITB           ;SETUP PORT A
4125  012554  112737  000101  056576          MOVB    #'A,MSG19A
4126  012562  013701  001342                  MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
4127  012566  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
4128  012572  001403                          BEQ     2$              ;YES, BRANCH
4129  012574  062737  000002  001222  1$:     ADD     #2,$UNIT        ;INCR TO NEXT EVEN DRIVE
4130  012602  005711                  2$:     TST     (R1)            ;IS DRIVE PRESENT?
4131  012604  001003                          BNE     5$              ;BR IF YES
4132  012606  062701  000004                  ADD     #4,R1           ;ELSE INCR PTR TO NEXT EVEN DRIVE
4133  012612  000770                          BR      1$              ;& TRY AGAIN
4134
4135  012614  005737  005426          5$:     TST     DDPCH           ;DDP CHAIN MODE?
4136  012620  001406                          BEQ     3$              ;BR IF NO
4137  012622  005737  001222                  TST     $UNIT           ;ELSE SEE IF DRV 0
4138  012626  001003                          BNE     3$              ;BR IF NO
4139  012630  062701  000004                  ADD     #4,R1           ;ELSE FIND NEXT EVEN DRIVE PRESENT
4140  012634  000757                          BR      1$
4141
4142  012636  032711  002000          3$:     BIT     #CDT,(R1)       ;SEE IF DRIVE UNDER TEST IS RK07
4143  012642  001403                          BEQ     6$              ;BR IF NO
4144  012644  012737  002000  001170          MOV     #CDT,$TMP4      ;ELSE SET RK07 FLAG
4145  012652  062701  000004          6$:     ADD     #4,R1
4146  012656  010137  001342                  MOV     R1,DRVPTR       ;STORE PTR OF NEXT EVEN DRIVE FLAG
4147  012662  104401  056500                  TYPE    ,MSG15          ;"DRIVE"
4148  012666  013700  001222                  MOV     $UNIT,R0
4149  012672  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
4150                                                                  ;;DRIVE #
4151  012674  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
4152  012676     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
4153  012677     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
4154
4155                          ;               TYPE    ,$CRLF  ;29-SEP-77
4156
4157
4158  012700  005737  001170                  TST     $TMP4           ;SEE IF RK07 UNDER TEST
4159  012704  001014                          BNE     7$              ;BR IF YES
4160  012706  012737  000632  012770          MOV     #632,LC         ;ELSE LOAD RK06 PARAMERERS
4161  012714  005037  012776                  CLR     E.DDT
4162  012720  012737  000777  012772          MOV     #777,MASK
4163  012726  012737  160017  012774          MOV     #160017,MASK1
4164  012734  000423                          BR      TST5            ;;GOTO NEXT TEST
4165
4166  012736  012737  001456  012770  7$:     MOV     #1456,LC        ;LOAD RK07 PARAMETERS
4167  012744  012737  000400  012776          MOV     #D.DDT,E.DDT
4168  012752  012737  001777  012772          MOV     #1777,MASK
4169  012760  012737  140017  012774          MOV     #140017,MASK1
```

```
                                        BR      TST5            ;;GOTO NEXT TEST

012770  000000                LC:     0                       ;LAST CYL
012772  000000                MASK:   0
012774  000000                MASK1:  0
012776  000000                E.DDT:  0                       ;EXPECTED DRIVE TYPE TO E.AD
013000  000000                FC:     0                       ;FIRST CYL
013002  000001                FCP1:   1                       ;FC+1
013004                      PFSRT:                            ;ENTER HERE FOR POWER FAIL RESTART

                        .SBTTL DUAL PORT TESTS

                        ;;*****************************************************************
                        ;*TEST 5         TEST PORT 'A' SEIZE & TIMEOUT
                        ;*
                        ;*      VERIFY THAT THE DRIVE CAN BE SEIZED & THAT THE PORT
                        ;*      TIMEOUT RELEASES THE DRIVE.
                        ;*
                        ;*      A.  SET VOLUME VALID FOR BOTH PORTS & DO RECAL COMMAND
                        ;*
                        ;*      B.  A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'A'.
                        ;*          THE PROGRAM VERIFIES THE DRIVE HAS BEEN SEIZED BY 'DRIVE
                        ;*          AVAILABLE' SET.
                        ;*
                        ;*      C.  A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'B' THE
                        ;*          PROGRAM VERIFIES THAT 'DRIVE AVAILABLE' IS NOT SET
                        ;*          FOR PORT 'B' & THAT CERR IS SET
                        ;*
                        ;*      D.  VERIFY THAT FOR ALL MESSAGES REQUESTED THRU PORT 'B', (MSG
                        ;*          A0-A3, B0-B3) THAT MESSAGE 0 ALWAYS RETURNS FROM PORT B
                        ;*          WHILE PORT 'A' IS SEIZED.
                        ;*
                        ;*      E.  WAIT FOR THE PORT TIMEOUT TO OCCUR ON PORT 'A' BY
                        ;*          CONTINUOUSLY CHECKING FOR ATTN ON PORT 'B'.
                        ;*          AFTER ATTN-B IS REC'D, A DRIVE SELECT COMMAND IS
                        ;*          ISSUED THRU PORT 'B' TO VERIFY THAT DRIVE AVAILABLE IS SET
                        ;*          IN MSG A0.
                        ;*
                        ;*          MEASURE THE DURATION OF THE TIMEOUT & TYPE THE VALUE
                        ;*          FOR THE FIRST PASS ONLY.
                        ;*
                        ;*      F.  VERIFY THAT ONLY PORT 'B' GETS 'DSC' & 'ATTN'.
                        ;*
                        ;*      G.  VERIFY THE DRIVE CLEAR COMMAND CLEARS 'DSC' & 'ATTN'
                        ;*          ON PORT 'B' BUT DOES NOT RELEASE THE DRIVE FROM PORT 'B'.
                        ;*
                        ;;*****************************************************************
013004  000004          TST5:   SCOPE
013006  012737  000001  001174  MOV     #1,$TIMES       ;;DO 1 ITERATION
013014  012706  001100          MOV     #STACK,SP
013020  012737  000000  005464  MOV     #0,UNITB        ;SETUP PORT A
013026  112737  000101  056576  MOVB    #'A,MSG19A
013034  013737  005466  001366  MOV     TIMER,COUNT
013042  004737  047204          JSR     PC,TMO          ;DO TIMEOUT

013046  004737  045534          JSR     PC,SUBCLR
```

```
4226  013052  104024                          ERROR   24              ;CERR AFTER SCLR
4227
4228
4229  013054  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
4230  013060  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TMO
4231  013062                          7$:
4232  013062  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4233  013070  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
4234  013076  063765  005464  000010          ADD     UNITB,RKCS2(R5)
4235  013104  012737  000003  005314          MOV     #PACK,HCS1
4236  013112  004737  043372                  JSR     PC,DOCMD        ;DO PACK CMD & GET CONTR RDY
4237  013116  104116                          ERROR   116             ;CONTR NOT RDY AFT PACK CMD
4238
4239  013120  032737  000100  005342          BIT     #D.VV,HMR2
4240  013126  001001                          BNE     64$
4241  013130  104027                          ERROR   27              ;VV NOT SET AFTER PACK CMD
4242  013132                          64$:
4243  013132  012737  042644  001176          MOV     #$EOP,$ESCAPE
4244  013140  005737  005464                  TST     UNITB
4245  013144  001022                          BNE     2$
4246  013146  012737  000001  005464          MOV     #1,UNITB                ;SETUP PORT B
4247  013154  112737  000102  056576          MOVB    #'B,MSG19A
4248  013162  013737  005466  001366          MOV     TIMER,COUNT
4249  013170  004737  047204                  JSR     PC.TMO          ;DO TIMEOUT
4250
4251  013174  004737  045534                  JSR     PC,SUBCLR
4252  013200  104223                          ERROR   223             ;CERR AFTER SCLR
4253
4254  013202  004737  044172                  JSR     PC,DRAV         ;SEE IF DROVE AVAIL
4255  013206  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TMO
4256  013210  000724                          BR      7$
4257  013212                          2$:
4258
4259  013212  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4260  013220  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
4261  013226  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
4262  013234  012737  000013  005314          MOV     #RECAL,HCS1
4263  013242  004737  043372                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
4264  013246  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
4265
4266  013250  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4267  013256  004737  045146                  JSR     PC,GSTAT
4268  013262  032737  020000  005342          BIT     #D.RTZ,HMR2
4269  013270  001001                          BNE     65$
4270  013272  104070                          ERROR   70              ;RTZ NOT SET DURING RECAL CMD
4271  013274  013737  001400  005354  65$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
4272  013302  004737  044006                  JSR     PC,FATT1        ;FIND ATTN
4273  013306  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
4274
4275  013310  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4276  013316  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4277  013322  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4278  013330  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4279  013336  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
4280  013342  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4281  013350  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
```

```
 4282
 4283  013356  004737  044274                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4284  013362  000007                            .WORD   T.A2!T.B2!T.B3          ;& MSGS SPECIFIED HERE
 4285  013364  104221                            ERROR   221             ;MSG A0 ERROR AFTER RECAL CMD
 4286  013366  104066                            ERROR   66              ;MSH B0 ERROR
 4287  013370  104222                            ERROR   222             ;MSG A1 ERROR
 4288  013372  104067                            ERROR   67              ;MSG B1 ERROR
 4289  013374  012765  000002  000026            MOV     #2,RKMR1(R5)
 4290  013402  004737  045146                    JSR     PC,GSTAT
 4291  013406  005737  001356                    TST     CYLDIF          ;SEE IF MSG A2=0
 4292  013412  001401                            BEQ     66$             ;BR IF YES
 4293  013414  104047                            ERROR   47              ;MSG A2 NOT CLEARED AFTER RECAL CMD
 4294  013416  005737  001360          66$:      TST     CYLADD          ;SEE IF MSG B2=0
 4295  013422  001401                            BEQ     67$             ;BR IF YES
 4296  013424  104050                            ERROR   50              ;MSG B2 NOT CLEARED AFTER RECAL CMD
 4297  013426                          67$:
 4298
 4299  013426  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
 4300  013434  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4301  013442  063765  005464  000010            ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
 4302  013450  012737  000005  005314            MOV     #CLEAR,HCS1
 4303  013456  004737  043372                    JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 4304  013462  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4305  013464  004737  043750                    JSR     PC.TSTATN       ;TEST FOR ATTN
 4306  013470  000401                            BR      68$
 4307  013472  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4308  013474                          68$:
 4309
 4310  013474  012737  010340  005404            MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 4311  013502  005037  005406                    CLR     E.B0            ;EXPECTED MSG B0
 4312  013506  012737  001720  005410            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4313  013514  012737  000001  005412            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4314  013522  005037  005414                    CLR     E.A2            ;EXPECTED MSG A2
 4315  013526  012737  000002  005416            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4316  013534  012737  000003  005422            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4317
 4318  013542  004737  044274                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4319  013546  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4320  013550  104033                            ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
 4321  013552  104034                            ERROR   34              ;MSH B0 ERROR
 4322  013554  104035                            ERROR   35              ;MSG A1 ERROR
 4323  013556  104036                            ERROR   36              ;MSG B1 ERROR
 4324
 4325
 4326  013560  012737  000000  005464            MOV     #0,UNITB        ;SETUP PORT A
 4327  013566  112737  000101  056576            MOVB    #'A,MSG19A
 4328  013574  013737  005466  001366            MOV     TIMER,COUNT
 4329  013602  004737  047204                    JSR     PC,TMO          ;DO TIMEOUT
 4330
 4331  013606  004737  045534                    JSR     PC,SUBCLR
 4332  013612  104024                            ERROR   24              ;CERR AFTER SCLR
 4333
 4334
 4335  013614  004737  044172                    JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
 4336  013620  104045                            ERROR   45              ;PORT A NOT AVAIL AFTER TMO
 4337  013622  012737  014544  001176            MOV     #5$,$ESCAPE
```

```
4338
4339  013630  012737  010340  005404        MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4340  013636  005037  005406                CLR     E.B0              ;EXPECTED MSG B0
4341  013642  012737  001720  005410        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4342  013650  012737  000001  005412        MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
4343  013656  005037  005414                CLR     E.A2              ;EXPECTED MSG A2
4344  013662  012737  000002  005416        MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
4345  013670  012737  000003  005422        MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
4346
4347  013676  004737  044274                JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
4348  013702  000000                        .WORD   0!0!0             ;& MSGS SPECIFIED HERE
4349  013704  104077                        ERROR   77                ;MSG A0 ERROR AFTER TIMEOUT
4350  013706  104100                        ERROR   100               ;MSH B0 ERROR
4351  013710  104101                        ERROR   101               ;MSG A1 ERROR
4352  013712  104102                        ERROR   102               ;MSG B1 ERROR
4353  013714  012737  000001  005464        MOV     #1,UNITB          ;SELECT PORT B BEFORE TIMEOUT OR RELEASE
4354  013722  112737  000102  056576        MOVB    #'B,MSG19A        ;SETUP ERROR MSG FOR PORT B
4355  013730  005037  001176                CLR     $ESCAPE
4356  013734  004737  044172                JSR     PC,DRAV           ;SEE IF DRIVE AVAIL
4357  013740  000401                        BR      1$                ;BR IF NOT AVAIL
4358  013742  104103                        ERROR   103               ;PORT B AVAIL BEFORE TMO OR RELEASE
4359
4360  013744  032737  100000  005314  1$:   BIT     #CERR,HCS1
4361  013752  001001                        BNE     6$
4362  013754  104130                        ERROR   130               ;CERR NOT SET AFTER SEL DRIVE & DRIVE NOT AVAIL
4363  013756  012737  010100  005404  6$:   MOV     #<D.SPIN!D.VV>,E.A0
4364  013764  013737  005404  005410        MOV     E.A0,E.A1         ;MSG 0 & 1 SHOULD ALWAYS RETURN SAME
4365  013772  053737  012776  005410        BIS     E.DDT,E.A1
4366  014000  005037  005406                CLR     E.B0              ;WORD 0 FOR PORT B
4367  014004  005037  005412                CLR     E.B1
4368  014010  012737  014544  001176        MOV     #5$,$ESCAPE
4369
4370  014016  004737  044274                JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
4371  014022  000000                        .WORD   0!0!0             ;& MSGS SPECIFIED HERE
4372  014024  104104                        ERROR   104               ;MSG A0 ERROR WHILE PORT B UNAVAILABLE
4373  014026  104105                        ERROR   105               ;MSH B0 ERROR
4374  014030  104106                        ERROR   106               ;MSG A1 ERROR
4375  014032  104107                        ERROR   107               ;MSG B1 ERROR
4376
4377  014034  005737  001216                TST     $PASS
4378  014040  001402                        BEQ     8$                ;BR IF FIRST PASS
4379  014042  000137  014544                JMP     5$                ;ELSE EXIT TEST
4380  014046  005037  001176          8$:   CLR     $ESCAPE
4381  014052  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
4382  014060  012737  000000  005464        MOV     #0,UNITB          ;SETUP FOR PORT A AGAIN
4383  014066  112737  000101  056576        MOVB    #'A,MSG19A
4384  014074  012737  000360  001366        MOV     #360,COUNT        ;SETUP 4 SEC TIMEOUT
4385  014102  004737  047114                JSR     PC,CLKON          ;TURN ON CLOCK
4386
4387  014106  004737  044172                JSR     PC,DRAV
4388  014112  104045                        ERROR   45                ;PORT A NOT AVAIL AFTER TIMEOUT
4389
4390  014114  012737  000001  005464        MOV     #1,UNITB          ;SELECT PORT B BEFORE TIMEOUT OR RELEASE
4391  014122  112737  000102  056576        MOVB    #'B,MSG19A
4392  014130  004737  044172                JSR     PC,DRAV           ;SEE IF PORT B DRIVE AVAIL
4393  014134  000401                        BR      3$                ;BR IF NOT AVAIL
```

```
014136  104103                            ERROR   103               ;PORT B AVAIL BEFORE TMO OR RELEASE

014140  012765  100000  000000  3$:       MOV     #CCLR,RKCS1(R5)
014146  013704  001222                    MOV     $UNIT,R4
014152  063704  005464                    ADD     UNITB,R4
014156  004737  044214                    JSR     PC,FATT3
014162  104110                            ERROR   110               ;NO ATTN ON PORT B TO ALLOW SEIZE

014164  004737  047162                    JSR     PC,CLKOF
014170  004737  044172                    JSR     PC,DRAV           ;SEE IF PORT B DRIVE AVAIL
014174  104045                            ERROR   45                ;PORT B NOT AVAIL

014176  012737  014544  001176            MOV     #5$,$ESCAPE

014204  012737  050340  005404            MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
014212  005037  005406                    CLR     E.B0              ;EXPECTED MSG B0
014216  012737  001720  005410            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
014224  012737  000001  005412            MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
014232  005037  005414                    CLR     E.A2              ;EXPECTED MSG A2
014236  012737  000002  005416            MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
014244  012737  000003  005422            MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3

014252  004737  044274                    JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
014256  000000                            .WORD   0!0!0             ;& MSGS SPECIFIED HERE
014260  104077                            ERROR   77                ;MSG A0 ERROR AFTER TIMEOUT
014262  104100                            ERROR   100               ;MSH B0 ERROR
014264  104101                            ERROR   101               ;MSG A1 ERROR
014266  104102                            ERROR   102               ;MSG B1 ERROR
014270  005037  001176                    CLR     $ESCAPE
014274  012737  000000  005464            MOV     #0,UNITB          ;SETUP FOR PORT A
014302  112737  000101  056576            MOVB    #'A,MSG19A
014310  004737  043750                    JSR     PC,TSTATN         ;TEST FOR ATTN ON PORT A
014314  000401                            BR      4$
014316  104111                            ERROR   111               ;PORT A ATTN SET W/O REQUEST PENDING

014320  012737  014544  001176  4$:       MOV     #5$,$ESCAPE
014326  012737  000001  005464            MOV     #1,UNITB          ;SETUP FOR PORT B
014334  112737  000102  056576            MOVB    #'B,MSG19A

014342  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
014350  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
014356  063765  005464  000010            ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
014364  012737  000005  005314            MOV     #CLEAR,HCS1
014372  004737  043372                    JSR     PC,DOCMD          ;DO DRIVE CLEAR CMD & GET CONTR RDY
014376  104151                            ERROR   151               ;NO RDY AFTER DRIVE CLEAR CMD
014400  004737  043750                    JSR     PC,TSTATN         ;TEST FOR ATTN
014404  000401                            BR      69$
014406  104154                            ERROR   154               ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
014410                          69$:

014410  012737  010340  005404            MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
014416  005037  005406                    CLR     E.B0              ;EXPECTED MSG B0
014422  012737  001720  005410            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
014430  012737  000001  005412            MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
014436  005037  005414                    CLR     E.A2              ;EXPECTED MSG A2
014442  012737  000002  005416            MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
```

```
4450  014450  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4451
4452  014456  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4453  014462  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4454  014464  104033                          ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
4455  014466  104034                          ERROR   34              ;MSH B0 ERROR
4456  014470  104035                          ERROR   35              ;MSG A1 ERROR
4457  014472  104036                          ERROR   36              ;MSG B1 ERROR
4458
4459
4460  014474  012701  000360                  MOV     #360,R1
4461  014500  163701  001366                  SUB     COUNT,R1        ;R1-COUNT=R1
4462  014504  012746  000021                  MOV     #17.,-(SP)              ;;PUT THE MULTIPLER ON THE STACK
4463  014510  010146                          MOV     R1,-(SP)                ;;PUT THE MULTIPLICAND ON THE STACK
4464  014512  004737  054270                  JSR     PC,@#$MULT      ;;CALL THE MULTPLY ROUTINE
4465  014516  012616                          MOV     (SP)+,(SP)      ;;DISREGARD THE MSB'S
4466  014520  012601                          MOV     (SP)+,R1                ;;GET THE LSB'S OF THE PRODUCT
4467  014522  104401  056602                  TYPE    ,MSG20          ;PORT TIMEOUT
4468  014526  010146                          MOV     R1,-(SP)        ;PUSH BINARY ONTO STACK
4469  014530  004737  054174                  JSR     PC,$SB2D        ;CONVERT TO ASCII
4470  014534  004737  054230                  JSR     PC,$SUPRS       ;TYPE IT
4471  014540  104401  056663                  TYPE    ,MSG22          ;MS
4472
4473  014544  005037  001176          55:     CLR     $ESCAPE
4474  014550  004737  047162                  JSR     PC,CLKOF
4475
4476                                  ;;**********************************************************************
4477                                  ;*TEST 6         TEST PORT 'B' SEIZE & TIMEOUT
4478                                  ;*
4479                                  ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
4480                                  ;*
4481                                  ;;**********************************************************************
4482  014554  000004                  TST6:   SCOPE
4483  014556  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4484  014564  012706  001100                  MOV     #STACK,SP
4485  014570  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
4486  014576  112737  000102  056576          MOVB    #'B,MSG19A
4487  014604  013737  005466  001366          MOV     TIMER,COUNT
4488  014612  004737  047204                  JSR     PC,TMO          ;DO TIMEOUT
4489
4490  014616  004737  045534                  JSR     PC,SUBCLR
4491  014622  104024                          ERROR   24              ;CERR AFTER SCLR
4492
4493
4494  014624  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
4495  014630  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TMO
4496  014632  012737  015554  001176          MOV     #55$,$ESCAPE
4497
4498  014640  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4499  014646  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4500  014652  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4501  014660  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4502  014666  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
4503  014672  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4504  014700  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4505
```

```
4506  014706  004737  044274          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4507  014712  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4508  014714  104077                  ERROR   77              ;MSG A0 ERROR AFTER TIMEOUT
4509  014716  104100                  ERROR   100             ;MSH B0 ERROR
4510  014720  104101                  ERROR   101             ;MSG A1 ERROR
4511  014722  104102                  ERROR   102             ;MSG B1 ERROR
4512  014724  012737  000000  005464  MOV     #0,UNITB        ;SELECT PORT A BEFORE TIMEOUT OR RELEASE
4513  014732  112737  000101  056576  MOVB    #'A,MSG19A      ;SETUP ERROR MSG FOR PORT A
4514  014740  005037  001176          CLR     $ESCAPE
4515  014744  004737  044172          JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
4516  014750  000401                  BR      1$              ;BR IF NOT AVAIL
4517  014752  104103                  ERROR   103             ;PORT A AVAIL BEFORE TMO OR RELEASE
4518
4519  014754  032737  100000  005314  1$:     BIT     #CERR,HCS1
4520  014762  001001                  BNE     6$
4521  014764  104130                  ERROR   130             ;CERR NOT SET AFTER SEL DRIVE & DRIVE NOT AVAIL
4522  014766  012737  010100  005404  6$:     MOV     #<D.SPIN!D.VV>,E.A0
4523  014774  013737  005404  005410          MOV     E.A0,E.A1       ;MSG 1 & 0 SHOULD ALWAYS RETURN SAME
4524  015002  053737  012776  005410          BIS     E.DDT,E.A1
4525  015010  005037  005406          CLR     E.B0            ;WORD 0 FOR PORT A
4526  015014  005037  005412          CLR     E.B1
4527  015020  012737  015554  001176  MOV     #5$,$ESCAPE
4528
4529  015026  004737  044274          JSR     PC.CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4530  015032  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4531  015034  104104                  ERROR   104             ;MSG A0 ERROR WHILE PORT A UNAVAILABLE
4532  015036  104105                  ERROR   105             ;MSH B0 ERROR
4533  015040  104106                  ERROR   106             ;MSG A1 ERROR
4534  015042  104107                  ERROR   107             ;MSG B1 ERROR
4535
4536  015044  005737  001216          TST     $PASS
4537  015050  001402                  BEQ     8$              ;BR IF FIRST PASS
4538  015052  000137  015554          JMP     5$              ;ELSE EXIT TEST
4539  015056  005037  001176  8$:     CLR     $ESCAPE
4540  015062  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
4541  015070  012737  000001  005464  MOV     #1,UNITB        ;SETUP FOR PORT B AGAIN
4542  015076  112737  000102  056576  MOVB    #'B,MSG19A
4543  015104  012737  000360  001366  MOV     #360,COUNT      ;SETUP 4 SEC TIMEOUT
4544  015112  004737  047114          JSR     PC,CLKON        ;TURN ON CLOCK
4545
4546  015116  004737  044172          JSR     PC,DRAV
4547  015122  104045                  ERROR   45              ;PORT B NOT AVAIL AFTER TIMEOUT
4548
4549  015124  012737  000000  005464  MOV     #0,UNITB        ;SELECT PORT A BEFORE TIMEOUT OR RELEASE
4550  015132  112737  000101  056576  MOVB    #'A,MSG19A
4551  015140  004737  044172          JSR     PC,DRAV         ;SEE IF PORT A DRIVE AVAIL
4552  015144  000401                  BR      3$              ;BR IF NOT AVAIL
4553  015146  104103                  ERROR   103             ;PORT A AVAIL BEFORE TMO OR RELEASE
4554
4555  015150  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
4556  015156  013704  001222          MOV     $UNIT,R4
4557  015162  063704  005464          ADD     UNITB,R4
4558  015166  004737  044214          JSR     PC,FATT3
4559  015172  104110                  ERROR   110             ;NO ATTN ON PORT A TO ALLOW SEIZE
4560
4561  015174  004737  047162          JSR     PC,CLKOF
```

```
4562  015200  004737  044172          JSR     PC,DRAV         ;SEE IF PORT A DRIVE AVAIL
4563  015204  104045                  ERROR   45              ;PORT A NOT AVAIL
4564
4565  015206  012737  015554  001176  MOV     #5$,$ESCAPE
4566
4567  015214  012737  050340  005404  MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4568  015222  005037  005406          CLR     E.B0            ;EXPECTED MSG B0
4569  015226  012737  001720  005410  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4570  015234  012737  000001  005412  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4571  015242  005037  005414          CLR     E.A2            ;EXPECTED MSG A2
4572  015246  012737  000002  005416  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4573  015254  012737  000003  005422  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4574
4575  015262  004737  044274          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4576  015266  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4577  015270  104077                  ERROR   77              ;MSG A0 ERROR AFTER TIMEOUT
4578  015272  104100                  ERROR   100             ;MSH B0 ERROR
4579  015274  104101                  ERROR   101             ;MSG A1 ERROR
4580  015276  104102                  ERROR   102             ;MSG B1 ERROR
4581  015300  005037  001176          CLR     $ESCAPE
4582  015304  012737  000001  005464  MOV     #1,UNITB        ;SETUP FOR PORT B
4583  015312  112737  000102  056576  MOVB    #'B,MSG19A
4584  015320  004737  043750          JSR     PC,TSTATN       ;TEST FOR ATTN ON PORT B
4585  015324  000401                  BR      4$
4586  015326  104111                  ERROR   111             ;PORT B ATTN SET W/O REQUEST PENDING
4587
4588  015330  012737  015554  001176  4$:     MOV     #5$,$ESCAPE
4589  015336  012737  000000  005464          MOV     #0,UNITB        ;SETUP FOR PORT A
4590  015344  112737  000101  056576          MOVB    #'A,MSG19A
4591
4592  015352  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4593  015360  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4594  015366  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
4595  015374  012737  000005  005314          MOV     #CLEAR,HCS1
4596  015402  004737  043372                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
4597  015406  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4598  015410  004737  043750                  JSR     PC,TSTATN       ;TEST FOR ATTN
4599  015414  000401                          BR      64$
4600  015416  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4601  015420                          64$:
4602
4603  015420  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4604  015426  005037  005406          CLR     E.B0            ;EXPECTED MSG B0
4605  015432  012737  001720  005410  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4606  015440  012737  000001  005412  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4607  015446  005037  005414          CLR     E.A2            ;EXPECTED MSG A2
4608  015452  012737  000002  005416  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4609  015460  012737  000003  005422  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4610
4611  015466  004737  044274          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4612  015472  000003                  .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4613  015474  104033                  ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
4614  015476  104034                  ERROR   34              ;MSH B0 ERROR
4615  015500  104035                  ERROR   35              ;MSG A1 ERROR
4616  015502  104036                  ERROR   36              ;MSG B1 ERROR
4617
```

```
4618
4619  015504  012701  000360                  MOV     #360,R1
4620  015510  163701  001366                  SUB     COUNT,R1        ;R1-COUNT=R1
4621  015514  012746  000021                  MOV     #17.,-(SP)              ;;PUT THE MULTIPLER ON THE STACK
4622  015520  010146                          MOV     R1,-(SP)                ;;PUT THE MULTIPLICAND ON THE STACK
4623  015522  004737  054270                  JSR     PC,@#$MULT      ;;CALL THE MULTPLY ROUTINE
4624  015526  012616                          MOV     (SP)+,(SP)      ;;DISREGARD THE MSB'S
4625  015530  012601                          MOV     (SP)+,R1                ;;GET THE LSB'S OF THE PRODUCT
4626  015532  104401  056667                  TYPE    ,MSG23          ;PORT TIMEOUT
4627  015536  010146                          MOV     R1,-(SP)        ;PUSH BINARY ONTO STACK
4628  015540  004737  054174                  JSR     PC,$SB2D        ;CONVERT TO ASCII
4629  015544  004737  054230                  JSR     PC,$SUPRS       ;TYPE IT
4630  015550  104401  056663                  TYPE    ,MSG22          ;MS
4631
4632  015554  005037  001176          5$:     CLR     $ESCAPE
4633  015560  004737  047162                  JSR     PC,CLKOF
4634
4635                                  ;;**********************************************************************
4636                                  ;*TEST 7         PRINT DRIVE SERIAL NUMBER
4637                                  ;*
4638                                  ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A3
4639                                  ;*      IN BCD ON THE 1'ST PASS ONLY.
4640                                  ;*      IT ALSO TESTS THAT THE SERIAL # READ THRU BOTH PORTS
4641                                  ;*      ARE THE SAME.
4642                                  ;*
4643                                  ;;**********************************************************************
4644  015564  000004                  TST7:   SCOPE
4645  015566  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 !TERATION
4646  015574  012706  001100                  MOV     #STACK,SP
4647
4648  015600  005737  001216                  TST     $PASS
4649  015604  001042                          BNE     TST10           ;;GO TO NEXT TST IF NOT 1ST PASS
4650
4651  015606  004737  045534                  JSR     PC,SUBCLR
4652  015612  104024                          ERROR   24              ;CERR AFTER SCLR
4653
4654  015614  104401  056512                  TYPE    ,MSG16          ;DRIVE SERIAL NO.
4655  015620  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
4656  015626  004737  045146                  JSR     PC,GSTAT        ;GET STATUS
4657  015632  013701  005342                  MOV     HMR2,R1         ;GET SERIAL #
4658  015636  012704  053762                  MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
4659  015642  010446                          MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
4660  015644  012703  000003                  MOV     #3,R3           ;SETUP CHAR COUNT
4661  015650  006101                          ROL     R1              ;INITIALIZE BIT POSITIONS
4662  015652  006101                          ROL     R1
4663  015654  006101                  1$:     ROL     R1              ;GET NEXT 4 BITS
4664  015656  006101                          ROL     R1
4665  015660  006101                          ROL     R1
4666  015662  006101                          ROL     R1
4667  015664  010100                          MOV     R1,R0           ;GET WORKING COPY
4668  015666  042700  177760                  BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
4669  015672  052700  000060                  BIS     #60,R0          ;CONVERT TO ASCII DIGIT
4670  015676  110024                          MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
4671  015700  005303                          DEC     R3
4672  015702  001364                          BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
4673  015704  105014                          CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR
```

```
015706  004737  054230                       JSR     PC,$SUPRS           ;TYPE
                                     ;       TYPE    ,$CRLF  ;29-SEP-77
                                     ;       TYPE    ,$CRLF  ;29-SEP-77

                                     ;;*******************************************************************
                                     ;*TEST 10        TEST PORT 'A' COMMAND SEIZE & ATTENTION
                                     ;*
                                     ;*       VERIFY THE OPERATION OF 'DSC' & 'ATTN' BITS AFTER A COMMAND.
                                     ;*
                                     ,*       A.   ISSUE A SEEK COMMAND TO CYLINDER 10 THRU PORT 'A'.
                                     ;*
                                     ;*       B.   VERIFY SEIZURE & THAT 'DSC' & 'ATTN' SETS FOR PORT 'A'
                                     ;*            ONLY AFTER SEEK COMPLETION
                                     ;*
                                     ;*       C.   VERIFY 'ATTN' REMAINS SET BEYOND TIMEOUT
                                     ;*
                                     ;*       D.   VERIFY A DRIVE CLEAR COMMAND RESETS 'DSC' & 'ATTN'
                                     ;*            & DOES NOT RELEASE THE DRIVE FROM PORT 'A'.
                                     ;*
                                     ;;*******************************************************************
015712  000004                       TST10:  SCOPE
015714  012737  000001  001174               MOV     #1,$TIMES           ;;DO 1 ITERATION
015722  012706  001100                       MOV     #STACK,SP
015726  012737  000000  005464               MOV     #0,UNITB            ;SETUP PORT A
015734  112737  000101  056576               MOVB    #'A,MSG19A
015742  013737  005466  001366               MOV     TIMER,COUNT
015750  004737  047204                       JSR     PC,TMO              ;DO TIMEOUT

015754  004737  045534                       JSR     PC,SUBCLR
015760  104024                               ERROR   24                  ;CERR AFTER SCLR


015762  004737  044172                       JSR     PC,DRAV             ;SEE IF DRIVE AVAIL
015766  104045                               ERROR   45                  ;PORT A NOT AVAIL AFTER TMO
015770  012765  000012  000020               MOV     #10.,RKDC(R5)       ;SEEK TO CYL 10.

015776  012737  000017  005314               MOV     #SEEK,HCS1
016004  004737  043372                       JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
016010  104131                               ERROR   131                 ;NO RDY AFTER SEEK CMD

016012  013737  001412  005352               MOV     T50000,TEMP1        ;SETUP TIMEOUT
016020  004737  044106                       JSR     PC,FATT2            ;FIND ATTN
016024  104132                               ERROR   132                 ;NO ATTN AFTER SEEK CMD

016026  032737  100000  005314               BIT     #CERR,HCS1
016034  001401                               BEQ     65$
016036  104210                               ERROR   210                 ,CERR AFTER SEEK CMD

016040                               65$:

016040  012737  050340  005404               MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
016046  005037  005406                       CLR     E.B0                ;EXPECTED MSG B0
016052  012737  001720  005410               MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
016060  012737  000001  005412               MOV     #1.E.B1             ;MSG ID FOR EXPECTED MSG B1
016066  005037  005414                       CLR     E.A2                ;EXPECTED MSG A2
```

```
 4730  016072  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4731  016100  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4732
 4733  016106  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4734  016112  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 4735  016114  104161                          ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
 4736  016116  104162                          ERROR   162             ;MSH B0 ERROR
 4737  016120  104163                          ERROR   163             ;MSG A1 ERROR
 4738  016122  104164                          ERROR   164             ;MSG B1 ERROR
 4739
 4740  016124  004737  045756                  JSR     PC,RDCYLA       ;READ CYL ADDR
 4741  016130  023727  001360  000012          CMP     CYLADD,#10.     ;SEE IF CYL 10.
 4742  016136  001415                          BEQ     64$             ;BR IF YES
 4743  016140  012737  000000  001344          MOV     #0,FRCYL
 4744  016146  012737  000012  001346          MOV     #10.,TOCYL
 4745  016154  012737  000012  001354          MOV     #10.,CALDIF
 4746  016162  012765  000012  000020          MOV     #10.,RKDC(R5)   ;REFRESH RKDC
 4747  016170  104224                          ERROR   224             ;DID NOT SEEK TO CYL 10.
 4748
 4749  016172  012737  000226  001366  64$:    MOV     #150.,COUNT
 4750  016200  004737  047204                  JSR     PC,TMO          ;DO 2.5 SEC TIMEOUT
 4751  016204  004737  044106                  JSR     PC,FATT2
 4752  016210  104112                          ERROR   112             ;ATTN CLEARED AFTER TMO
 4753
 4754  016212  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4755  016220  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 4756  016226  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
 4757  016234  012737  000005  005314          MOV     #CLEAR,HCS1
 4758  016242  004737  043372                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 4759  016246  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 4760  016250  004737  043750                  JSR     PC,TSTATN       ;TEST FOR ATTN
 4761  016254  000401                          BR      66$
 4762  016256  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4763  016260                          66$:
 4764
 4765  016260  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 4766  016266  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
 4767  016272  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4768  016300  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4769  016306  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
 4770  016312  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4771  016320  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4772
 4773  016326  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4774  016332  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4775  016334  104033                          ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
 4776  016336  104034                          ERROR   34              ;MSH B0 ERROR
 4777  016340  104035                          ERROR   35              ;MSG A1 ERROR
 4778  016342  104036                          ERROR   36              ;MSG B1 ERROR
 4779
 4780  016344  004737  044172                  JSR     PC,DRAV
 4781  016350  104113                          ERROR   113             ;PORT A NOT AVAIL AFTER DRIVE CLEAR CMD
 4782
 4783                          ;;*************************************************************
 4784                          ;*TEST 11        TEST PORT 'B' COMMAND SEIZE & ATTENTION
 4785                          ;*
```

```
                                        ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B',
                                        ;*      BUT THE SEEK IS TO CYLINDER 0
                                        ;*
                                        ;;**********************************************************************
016352  000004                  TST11:  SCOPE
016354  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
016362  012706  001100                  MOV     #STACK,SP
016366  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
016374  112737  000102  056576          MOVB    #'B,MSG19A
016402  013737  005466  001366          MOV     TIMER,COUNT
016410  004737  047204                  JSR     PC,TMO          ;DO TIMEOUT

016414  004737  045534                  JSR     PC,SUBCLR
016420  104024                          ERROR   24              ;CERR AFTER SCLR


016422  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
016426  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TMO
016430  012765  000000  000020          MOV     #0,RKDC(R5)     ;SEEK TO CYL 0

016436  012737  000017  005314          MOV     #SEEK,HCS1
016444  004737  043372                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
016450  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD

016452  013737  001412  005352          MOV     T50000,TEMP1    ;SETUP TIMEOUT
016460  004737  044106                  JSR     PC,FATT2        ;FIND ATTN
016464  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD

016466  032737  100000  005314          BIT     #CERR,HCS1
016474  001401                          BEQ     65$
016476  104210                          ERROR   210             ;CERR AFTER SEEK CMD

016500                          65$:

016500  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
016506  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
016512  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
016520  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
016526  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
016532  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
016540  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

016546  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
016552  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
016554  104161                          ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
016556  104162                          ERROR   162             ;MSH B0 ERROR
016560  104163                          ERROR   163             ;MSG A1 ERROR
016562  104164                          ERROR   164             ;MSG B1 ERROR

016564  004737  045756                  JSR     PC,RDCYLA       ;READ CYL ADDR
016570  023727  001360  000000          CMP     CYLADD,#0       ;SEE IF CYL 0
016576  001415                          BEQ     64$             ;BR IF YES
016600  012737  000012  001344          MOV     #10.,FRCYL
016606  012737  000000  001346          MOV     #0,TOCYL
016614  012737  000000  001354          MOV     #0,CALDIF
016622  012765  000000  000020          MOV     #0,RKDC(R5)     ;REFRESH RKDC
```

```
4842  016630  104224                                  ERROR   224             ;DID NOT SEEK TO CYL 0
4843
4844  016632  012737  000226  001366  64$:    MOV     #150.,COUNT
4845  016640  004737  047204                  JSR     PC,TMO          ;DO 2.5 SEC TIMEOUT
4846  016644  004737  044106                  JSR     PC,FATT2
4847  016650  104112                          ERROR   112             ;ATTN CLEARED AFTER TMO
4848
4849  016652  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4850  016660  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4851  016666  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
4852  016674  012737  000005  005314          MOV     #CLEAR,HCS1
4853  016702  004737  043372                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
4854  016706  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4855  016710  004737  043750                  JSR     PC,TSTATN       ;TEST FOR ATTN
4856  016714  000401                          BR      66$
4857  016716  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4858  016720                          66$:
4859
4860  016720  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4861  016726  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4862  016732  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4863  016740  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4864  016746  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
4865  016752  012737  000002  005416          MOV     #2.E.B2         ;MSG ID FOR EXPECTED MSG B2
4866  016760  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4867
4868  016766  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4869  016772  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4870  016774  104033                          ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
4871  016776  104034                          ERROR   34              ;MSH B0 ERROR
4872  017000  104035                          ERROR   35              ;MSG A1 ERROR
4873  017002  104036                          ERROR   36              ;MSG B1 ERROR
4874
4875  017004  004737  044172                  JSR     PC,DRAV
4876  017010  104113                          ERROR   113             ;PORT B NOT AVAIL AFTER DRIVE CLEAR CMD
4877
4878                                  ;;**********************************************************************
4879                                  ;*TEST 12       TEST RESET PORT 'A' ATTENTION BY DRIVE CLEAR COMMAND
4880                                  ;*
4881                                  ;*      VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF
4882                                  ;*      THE SEIZING PORT
4883                                  ;*
4884                                  ;*      A.  SET EACH PORT'S ATTENTION BIT BY PERFORMING SEEK
4885                                  ;*          COMMANDS TO CYLINDER 0 & ALLOWING TIMEOUTS.
4886                                  ;*
4887                                  ;*      B.  SEIZE THE DRIVE THRU PORT 'A' & ISSUE A DRIVE CLEAR COMMAND
4888                                  ;*          VERIFY THAT 'DSC' & 'ATTN' FOR PORT 'A' HAVE BEEN CLEARED
4889                                  ;*
4890                                  ;*      C.  SEIZE THE DRIVE THRU PORT 'B' & VERIFY 'DSC' & 'ATTN'
4891                                  ;*          HAVE NOT CLEARED
4892                                  ;;**********************************************************************
4893  017012  000004                  TST12:  SCOPE
4894  017014  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4895  017022  012706  001100                  MOV     #STACK,SP
4896  017026  012737  000000  005464          MOV     #0,UNITB        ;SETUP PORT A
4897  017034  112737  000101  056576          MOVB    #'A,MSG19A
```

```
4898  017042  013737  005466  001366          MOV     TIMER,COUNT
4899  017050  004737  047204                  JSR     PC,TMO          ;DO TIMEOUT
4900
4901  017054  004737  045534                  JSR     PC,SUBCLR
4902  017060  104024                          ERROR   24              ;CERR AFTER SCLR
4903
4904
4905  017062  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
4906  017066  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TMO
4907
4908  017070  012737  000017  005314          MOV     #SEEK,HCS1
4909  017076  004737  043372                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
4910  017102  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
4911
4912  017104  013737  001412  005352          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4913  017112  004737  044106                  JSR     PC,FATT2        ;FIND ATTN
4914  017116  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
4915
4916  017120  032737  100000  005314          BIT     #CERR,HCS1
4917  017126  001401                          BEQ     64$
4918  017130  104210                          ERROR   210             ;CERR AFTER SEEK CMD
4919
4920  017132                          64$:
4921
4922  017132  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4923  017140  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4924  017144  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4925  017152  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4926  017160  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
4927  017164  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4928  017172  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4929
4930  017200  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4931  017204  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4932  017206  104161                          ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
4933  017210  104162                          ERROR   162             ;MSH B0 ERROR
4934  017212  104163                          ERROR   163             ;MSG A1 ERROR
4935  017214  104164                          ERROR   164             ;MSG B1 ERROR
4936
4937  017216  013737  005466  001366          MOV     TIMER,COUNT
4938  017224  004737  047204                  JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT
4939  017230  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
4940  017236  112737  000102  056576          MOVB    #'B,MSG19A
4941  017244  004737  044172                  JSR     PC,DRAV
4942  017250  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TMO
4943
4944
4945  017252  012737  010340  005404          MOV     #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4946  017260  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4947  017264  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4948  017272  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4949  017300  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
4950  017304  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4951  017312  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4952
4953  017320  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
```

```
4954  017324  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4955  017326  104165                          ERROR   165             ;MSG A0 ERROR AFTER TIMEOUT
4956  017330  104166                          ERROR   166             ;MSH B0 ERROR
4957  017332  104167                          ERROR   167             ;MSG A1 ERROR
4958  017334  104170                          ERROR   170             ;MSG B1 ERROR
4959
4960  017336  012737  000017  005314          MOV     #SEEK,HCS1
4961  017344  004737  043372                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
4962  017350  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
4963
4964  017352  013737  001412  005352          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4965  017360  004737  044106                  JSR     PC,FATT2        ;FIND ATTN
4966  017364  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
4967
4968  017366  032737  100000  005314          BIT     #CERR,HCS1
4969  017374  001401                          BEQ     65$
4970  017376  104210                          ERROR   210             ;CERR AFTER SEEK CMD
4971
4972  017400                          65$:
4973
4974  017400  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4975  017406  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4976  017412  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4977  017420  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4978  017426  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
4979  017432  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4980  017440  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4981
4982  017446  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4983  017452  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
4984  017454  104161                          ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
4985  017456  104162                          ERROR   162             ;MSH B0 ERROR
4986  017460  104163                          ERROR   163             ;MSG A1 ERROR
4987  017462  104164                          ERROR   164             ;MSG B1 ERROR
4988
4989  017464  013737  005466  001366          MOV     TIMER,COUNT
4990  017472  004737  047204                  JSR     PC,TMO
4991  017476  012737  000000  005464          MOV     #0,UNITB        ;SETUP PORTC
4992  017504  112737  000101  056576          MOVB    #'A,MSG19A
4993  017512  004737  044172                  JSR     PC,DRAV
4994  017516  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TMO
4995
4996
4997  017520  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
4998  017526  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
4999  017532  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ,EXPECTED A1
5000  017540  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5001  017546  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5002  017552  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5003  017560  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5004
5005  017566  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5006  017572  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5007  017574  104165                          ERROR   165             ;MSG A0 ERROR AFTER TIMEOUT
5008  017576  104166                          ERROR   166             ;MSH B0 ERROR
5009  017600  104167                          ERROR   167             ;MSG A1 ERROR
```

```
5010  017602  104170                          ERROR   170             ;MSG B1 ERROR
5011
5012  017604  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5013  017612  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5014  017620  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
5015  017626  012737  000005  005314          MOV     #CLEAR,HCS1
5016  017634  004737  043372                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5017  017640  104151                          ERROR   151             ;NC RDY AFTER DRIVE CLEAR CMD
5018  017642  004737  043750                  JSR     PC,TSTATN       ;TEST FOR ATTN
5019  017646  000401                          BR      66$
5020  017650  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5021  017652                          66$:
5022
5023  017652  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5024  017660  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5025  017664  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5026  017672  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5027  017700  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5028  017704  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5029  017712  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5030
5031  017720  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5032  017724  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5033  017726  104033                          ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
5034  017730  104034                          ERROR   34              ;MSH B0 ERROR
5035  017732  104035                          ERROR   35              ;MSG A1 ERROR
5036  017734  104036                          ERROR   36              ;MSG B1 ERROR
5037
5038  017736  013737  005466  001366          MOV     TIMER,COUNT
5039  017744  004737  047204                  JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT
5040  017750  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
5041  017756  112737  000102  056576          MOVB    #'B,MSG19A
5042  017764  004737  043750                  JSR     PC,TSTATN
5043  017770  104114                          ERROR   114             ;ATTN RESET ON PORT B AFTER DR CLR CMD
5044                                                                  ;ON PORT A
5045
5046
5047  017772  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5048  020000  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5049  020004  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOP!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5050  020012  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5051  020020  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5052  020024  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5053  020032  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5054
5055  020040  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5056  020044  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5057  020046  104165                          ERROR   165             ;MSG A0 ERROR AFTER TIMEOUT
5058  020050  104166                          ERROR   166             ;MSH B0 ERROR
5059  020052  104167                          ERROR   167             ;MSG A1 ERROR
5060  020054  104170                          ERROR   170             ;MSG B1 ERROR
5061
5062                          ;;*****************************************************************
5063                          ;*TEST 13       TEST RESET PORT 'B' ATTENTION BY DRIVE CLEAR COMMAND
5064                          ;*
5065                          ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
```

```
5066                                        ;*
5067                                        ;;**********************************************************************
5068  020056  000004                        TST13:  SCOPE
5069  020060  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
5070  020066  012706  001100                        MOV     #STACK,SP
5071  020072  012737  000001  005464                MOV     #1,UNITB        ;SETUP PORT B
5072  020100  112737  000102  056576                MOVB    #'B,MSG19A
5073  020106  013737  005466  001366                MOV     TIMER,COUNT
5074  020114  004737  047204                        JSR     PC,TMO          ;DO TIMEOUT
5075
5076  020120  004737  045534                        JSR     PC,SUBCLR
5077  020124  104024                                ERROR   24              ;CERR AFTER SCLR
5078
5079
5080  020126  004737  044172                        JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
5081  020132  104045                                ERROR   45              ;PORT B NOT AVAIL AFTER TMO
5082
5083  020134  012737  000017  005314                MOV     #SEEK,HCS1
5084  020142  004737  043372                        JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
5085  020146  104131                                ERROR   131             ;NO RDY AFTER SEEK CMD
5086
5087  020150  013737  001412  005352                MOV     T50000,TEMP1    ;SETUP TIMEOUT
5088  020156  004737  044106                        JSR     PC,FATT2        ;FIND ATTN
5089  020162  104132                                ERROR   132             ;NO ATTN AFTER SEEK CMD
5090
5091  020164  032737  100000  005314                BIT     #CERR,HCS1
5092  020172  001401                                BEQ     64$
5093  020174  104210                                ERROR   210             ;CERR AFTER SEEK CMD
5094
5095  020176                                64$:
5096
5097  020176  012737  050340  005404                MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5098  020204  005037  005406                        CLR     E.B0            ;EXPECTED MSG B0
5099  020210  012737  001720  005410                MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5100  020216  012737  000001  005412                MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5101  020224  005037  005414                        CLR     E.A2            ;EXPECTED MSG A2
5102  020230  012737  000002  005416                MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5103  020236  012737  000003  005422                MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5104
5105  020244  004737  044274                        JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5106  020250  000000                                .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5107  020252  104161                                ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
5108  020254  104162                                ERROR   162             ;MSH B0 ERROR
5109  020256  104163                                ERROR   163             ;MSG A1 ERROR
5110  020260  104164                                ERROR   164             ;MSG B1 ERROR
5111
5112  020262  013737  005466  001366                MOV     TIMER,COUNT
5113  020270  004737  047204                        JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT
5114  020274  012737  000000  005464                MOV     #0,UNITB        ;SETUP PORT A
5115  020302  112737  000101  056576                MOVB    #'A,MSG19A
5116  020310  004737  044172                        JSR     PC,DRAV
5117  020314  104045                                ERROR   45              ;PORT A NOT AVAIL AFTER TMO
5118
5119
5120  020316  012737  010340  005404                MOV     #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5121  020324  005037  005406                        CLR     E.B0            ;EXPECTED MSG B0
```

```
5122  020330  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5123  020336  012737  000001  005412          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5124  020344  005037  005414                  CLR     E.A2                ;EXPECTED MSG A2
5125  020350  012737  000002  005416          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5126  020356  012737  000003  005422          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5127
5128  020364  004737  044274                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5129  020370  000000                          .WORD   0!0!0               ;& MSGS SPECIFIED HERE
5130  020372  104165                          ERROR   165                 ;MSG A0 ERROR AFTER TIMEOUT
5131  020374  104166                          ERROR   166                 ;MSH B0 ERROR
5132  020376  104167                          ERROR   167                 ;MSG A1 ERROR
5133  020400  104170                          ERROR   170                 ;MSG B1 ERROR
5134
5135  020402  012737  000017  005314          MOV     #SEEK,HCS1
5136  020410  004737  043372                  JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
5137  020414  104131                          ERROR   131                 ;NO RDY AFTER SEEK CMD
5138
5139  020416  013737  001412  005352          MOV     T50000,TEMP1        ;SETUP TIMEOUT
5140  020424  004737  044106                  JSR     PC,FATT2            ;FIND ATTN
5141  020430  104132                          ERROR   132                 ;NO ATTN AFTER SEEK CMD
5142
5143  020432  032737  100000  005314          BIT     #CERR,HCS1
5144  020440  001401                          BEQ     65$
5145  020442  104210                          ERROR   210                 ;CERR AFTER SEEK CMD
5146
5147  020444                          65$:
5148
5149  020444  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5150  020452  005037  005406                  CLR     E.B0                ;EXPECTED MSG B0
5151  020456  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5152  020464  012737  000001  005412          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5153  020472  005037  005414                  CLR     E.A2                ;EXPECTED MSG A2
5154  020476  012737  000002  005416          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5155  020504  012737  000003  005422          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5156
5157  020512  004737  044274                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5158  020516  000000                          .WORD   0!0!0               ;& MSGS SPECIFIED HERE
5159  020520  104161                          ERROR   161                 ;MSG A0 ERROR AFTER SEEK CMD
5160  020522  104162                          ERROR   162                 ;MSH B0 ERROR
5161  020524  104163                          ERROR   163                 ;MSG A1 ERROR
5162  020526  104164                          ERROR   164                 ;MSG B1 ERROR
5163
5164  020530  013737  005466  001366          MOV     TIMER,COUNT
5165  020536  004737  047204                  JSR     PC,TM0
5166  020542  012737  000001  005464          MOV     #1,UNITB            ;SETUP PORTC
5167  020550  112737  000102  056576          MOVB    #'B,MSG19A
5168  020556  004737  044172                  JSR     PC,DRAV
5169  020562  104045                          ERROR   45                  ;PORT B NOT AVAIL AFTER TM0
5170
5171
5172  020564  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5173  020572  005037  005406                  CLR     E.B0                ;EXPECTED MSG B0
5174  020576  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5175  020604  012737  000001  005412          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5176  020612  005037  005414                  CLR     E.A2                ;EXPECTED MSG A2
5177  020616  012737  000002  005416          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
```

```
 5178  020624  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 5179
 5180  020632  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 5181  020636  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
 5182  020640  104165                          ERROR   165              ;MSG A0 ERROR AFTER TIMEOUT
 5183  020642  104166                          ERROR   166              ;MSH B0 ERROR
 5184  020644  104167                          ERROR   167              ;MSG A1 ERROR
 5185  020646  104170                          ERROR   170              ;MSG B1 ERROR
 5186
 5187  020650  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 5188  020656  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 5189  020664  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
 5190  020672  012737  000005  005314          MOV     #CLEAR,HCS1
 5191  020700  004737  043372                  JSR     PC,DOCMD         ;DO DRIVE CLEAR CMD & GET CONTR RDY
 5192  020704  104151                          ERROR   151              ;NO RDY AFTER DRIVE CLEAR CMD
 5193  020706  004737  043750                  JSR     PC,TSTATN        ;TEST FOR ATTN
 5194  020712  000401                          BR      66$
 5195  020714  104154                          ERROR   154              ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 5196  020716                          66$:
 5197
 5198  020716  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 5199  020724  005037  005406                  CLR     E.B0             ;EXPECTED MSG B0
 5200  020730  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 5201  020736  012737  000001  005412          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
 5202  020744  005037  005414                  CLR     E.A2             ;EXPECTED MSG A2
 5203  020750  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 5204  020756  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 5205
 5206  020764  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 5207  020770  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
 5208  020772  104033                          ERROR   33               ;MSG A0 ERROR AFTER DRV CLEAR CMD
 5209  020774  104034                          ERROR   34               ;MSH B0 ERROR
 5210  020776  104035                          ERROR   35               ;MSG A1 ERROR
 5211  021000  104036                          ERROR   36               ;MSG B1 ERROR
 5212
 5213  021002  013737  005466  001366          MOV     TIMER,COUNT
 5214  021010  004737  047204                  JSR     PC,TM0           ;DO 1.5 SEC TIMEOUT
 5215  021014  012737  000000  005464          MOV     #0,UNITB         ;SETUP PORT A
 5216  021022  112737  000101  056576          MOVB    #'A,MSG19A
 5217  021030  004737  043750                  JSR     PC,TSTATN
 5218  021034  104114                          ERROR   114              ;ATTN RESET ON PORT A AFTER DR CLR CMD
 5219                                           ;ON PORT B
 5220
 5221
 5222  021036  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 5223  021044  005037  005406                  CLR     E.B0             ;EXPECTED MSG B0
 5224  021050  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 5225  021056  012737  000001  005412          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
 5226  021064  005037  005414                  CLR     E.A2             ;EXPECTED MSG A2
 5227  021070  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 5228  021076  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 5229
 5230  021104  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 5231  021110  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
 5232  021112  104165                          ERROR   165              ;MSG A0 ERROR AFTER TIMEOUT
 5233  021114  104166                          ERROR   166              ;MSH B0 ERROR
```

```
                                                ERROR   167             ;MSG A1 ERROR
                                                ERROR   170             ;MSG B1 ERROR

                                        ;;**********************************************************************
                                        ;*TEST 14       TEST RELEASE, DRIVE SEIZED BY PORT 'A'
                                        ;*
                                        ;*      A.  SEIZE THE DRIVE THRU PORT 'A'
                                        ;*
                                        ;*      B.  ISSUE A RELEASE USING RKCS2 THRU PORT 'A'
                                        ;*
                                        ;*      C.  VERIFY PORT 'B' CAN ACCESS THE DRIVE IMMEDIATELY &
                                        ;*          THAT NEITHER PORT SEES 'DSC' OR 'ATTN'
                                        ;*
                                        ;;**********************************************************************
021122  000004                  TST14:  SCOPE
021124  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
021132  012706  001100                  MOV     #STACK,SP
021136  012737  000000  005464          MOV     #0,UNITB        ;SETUP PORT A
021144  112737  000101  056576          MOVB    #'A,MSG19A
021152  013737  005466  001366          MOV     TIMER,COUNT
021160  004737  047204                  JSR     PC,TMO          ;DO TIMEOUT

021164  004737  045534                  JSR     PC,SUBCLR
021170  104024                          ERROR   24              ;CERR AFTER SCLR


021172  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
021176  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TMO
021200  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
021206  012737  000000  005464          MOV     #0,UNITB
021214  063765  005464  000010          ADD     UNITB,RKCS2(R5)
021222  112737  000101  056576          MOVB    #'A,MSG19A
021230  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
021236  012737  000001  005314          MOV     #SELDRV,HCS1
021244  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
021250  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD

021252  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
021260  012737  000001  005464          MOV     #1,UNITB
021266  063765  005464  000010          ADD     UNITB,RKCS2(R5)
021274  112737  000102  056576          MOVB    #'B,MSG19A
021302  012737  000001  005314          MOV     #SELDRV,HCS1
021310  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
021314  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD

021316  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
021324  001001                          BNE     64$             ;BR IF YES
021326  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
021330                          64$:
021330  004737  043750                  JSR     PC,TSTATN
021334  000401                          BR      1$
021336  104115                          ERROR   115             ;ATTN SET IN PORT B AFTER RLS OF PORT A
021340                          1$:

021340  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
021346  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
```

```
5290  021352  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5291  021360  012737  000001  005412          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
5292  021366  005037  005414                  CLR     E.A2             ;EXPECTED MSG A2
5293  021372  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
5294  021400  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
5295
5296  021406  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
5297  021412  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
5298  021414  104133                          ERROR   133              ;MSG A0 ERROR AFTER RELEASED TO PORT B
5299  021416  104134                          ERROR   134              ;MSH B0 ERROR
5300  021420  104135                          ERROR   135              ;MSG A1 ERROR
5301  021422  104136                          ERROR   136              ;MSG B1 ERROR
5302
5303  021424  012737  000000  005464          MOV     #0,UNITB         ;ADDRESS PORT A
5304  021432  112737  000101  056576          MOVB    #'A,MSG19A
5305  021440  004737  043750                  JSR     PC,TSTATN
5306  021444  000433                          BR      TST15            ;;GOTO NEXT TST
5307  021446  104115                          ERROR   115              ;ATTN SET AFTER RLS ISSUED
5308
5309  021450  012737  010340  005404          MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5310  021456  005037  005406                  CLR     E.B0             ;EXPECTED MSG B0
5311  021462  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5312  021470  012737  000001  005412          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
5313  021476  005037  005414                  CLR     E.A2             ;EXPECTED MSG A2
5314  021502  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
5315  021510  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
5316
5317  021516  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
5318  021522  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
5319  021524  104211                          ERROR   211              ;MSG A0 ERROR AFTER RELEASE ISSUED
5320  021526  104212                          ERROR   212              ;MSH B0 ERROR
5321  021530  104213                          ERROR   213              ;MSG A1 ERROR
5322  021532  104214                          ERROR   214              ;MSG B1 ERROR
5323
5324                                  ;;*************************************************************
5325                                  ;*TEST 15       TEST RELEASE, DRIVE SEIZED BY PORT 'B'
5326                                  ;*
5327                                  ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
5328                                  ;*
5329                                  ;;*************************************************************
5330  021534  000004                  TST15:  SCOPE
5331  021536  012737  000001  001174          MOV     #1,$TIMES        ;;DO 1 ITERATION
5332  021544  012706  001100                  MOV     #STACK,SP
5333  021550  012737  000001  005464          MOV     #1,UNITB         ;SETUP PORT B
5334  021556  112737  000102  056576          MOVB    #'B,MSG19A
5335  021564  013737  005466  001366          MOV     TIMER,COUNT
5336  021572  004737  047204                  JSR     PC,TMO           ;DO TIMEOUT
5337
5338  021576  004737  045534                  JSR     PC,SUBCLR
5339  021602  104024                          ERROR   24               ;CERR AFTER SCLR
5340
5341
5342  021604  004737  044172                  JSR     PC,DRAV          ;SEE IF DRIVE AVAIL
5343  021610  104045                          ERROR   45               ;PORT B NOT AVAIL AFTER TMO
5344  021612  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5345  021620  012737  000001  005464          MOV     #1,UNITB
```

```
5346  021626  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5347  021634  112737  000102  056576          MOVB    #'B,MSG19A
5348  021642  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
5349  021650  012737  000001  005314          MOV     #SELDRV,HCS1
5350  021656  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5351  021662  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5352
5353  021664  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5354  021672  012737  000000  005464          MOV     #0,UNITB
5355  021700  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5356  021706  112737  000101  056576          MOVB    #'A,MSG19A
5357  021714  012737  000001  005314          MOV     #SELDRV,HCS1
5358  021722  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5359  021726  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5360
5361  021730  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
5362  021736  001001                          BNE     64$             ;BR IF YES
5363  021740  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
5364  021742                          64$:
5365  021742  004737  043750                  JSR     PC,TSTATN
5366  021746  000401                          BR      1$
5367  021750  104115                          ERROR   115             ;ATTN SET IN PORT A AFTER RLS OF PORT B
5368  021752                          1$:
5369
5370  021752  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5371  021760  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5372  021764  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5373  021772  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5374  022000  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5375  022004  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5376  022012  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5377
5378  022020  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5379  022024  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5380  022026  104133                          ERROR   133             ;MSG A0 ERROR AFTER RELEASED TO PORT A
5381  022030  104134                          ERROR   134             ;MSH B0 ERROR
5382  022032  104135                          ERROR   135             ;MSG A1 ERROR
5383  022034  104136                          ERROR   136             ;MSG B1 ERROR
5384
5385  022036  012737  000001  005464          MOV     #1,UNITB        ;ADDRESS PORT B
5386  022044  112737  000102  056576          MOVB    #'B,MSG19A
5387  022052  004737  043750                  JSR     PC,TSTATN
5388  022056  000433                          BR      TST16           ;;GOTO NEXT TST
5389  022060  104115                          ERROR   115             ;ATTN SET AFTER RLS ISSUED
5390
5391  022062  012737  010340  005404          MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5392  022070  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5393  022074  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5394  022102  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5395  022110  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5396  022114  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5397  022122  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5398
5399  022130  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5400  022134  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5401  022136  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
```

```
                                ERROR   212             ;MSH B0 ERROR
                                ERROR   213             ;MSG A1 ERROR
                                ERROR   214             ;MSG B1 ERROR

                        ;;**********************************************************************
                        ;*TEST 16       TEST RELEASE FROM PORT 'A' WITH PORT 'B' REQUESTING
                        ;*
                        ;*      A.  PORT 'A' SEIZES THE DRIVE & DOES A SEEK TO SELF COMMAND.
                        ;*          THE PROGRAM VERIFIES 'DSC' & 'ATTN' ON PORT 'A' ONLY
                        ;*          ON COMPLETION
                        ;*
                        ;*      B.  PORT 'B' TRIES TO ACCESS THE DRIVE.   THE PROGRAM VERIFIES
                        ;*          DRIVE NOT AVAILABLE
                        ;*
                        ;*      C.  A RELEASE BY PORT 'A' IS ISSUED.  VERIFY PORT 'B' CAN
                        ;*          ACCESS THE DRIVE IMMEDIATELY & THAT 'DSC' & 'ATTN'
                        ;*          ARE SEEN ON PORT 'B'
                        ;*
                        ;*      D.  VERIFY PORT 'A' 'DSC' & 'ATTN' REMAINS SET AFTER RELEASE
                        ;*
                        ;*      E.  THE PROGRAM ISSUES A DRIVE CLEAR COMMAND TO PORT 'B'
                        ;*          & VERIFIES 'DSC' & 'ATTN' RESETS.
                        ;*
                        ;*      F.  THE PROGRAM THEN VERIFIES THAT PORT 'B' DOES NOT SEE
                        ;*          FURTHER (MULTIPLE) ATTENTIONS FROM WHAT WOULD HAVE BEEN
                        ;*          NORMAL TIMEOUT FROM PORT 'A'.
                        ;*
                        ;;**********************************************************************
                TST16:  SCOPE
                        MOV     #1,$TIMES       ;;DO 1 ITERATION
                        MOV     #STACK,SP
                        MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
                        MOV     #1,UNITB
                        ADD     UNITB,RKCS2(R5)
                        MOVB    #'B,MSG19A
                        ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
                        MOV     #SELDRV,HCS1
                        JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
                        ERROR   117             ;NO RDY AFTER SEL DRV CMD

                        MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
                        MOV     #0,UNITB
                        ADD     UNITB,RKCS2(R5)
                        MOVB    #'A,MSG19A
                        MOV     #SELDRV,HCS1
                        JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
                        ERROR   117             ;NO RDY AFTER SEL DRV CMD

                        BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
                        BNE     64$             ;BR IF YES
                        ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
                64$:

                        MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
                        CLR     E.B0            ;EXPECTED MSG B0
                        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
```

```
5458  022332  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5459  022340  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5460  022344  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5461  022352  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5462
5463  022360  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5464  022364  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5465  022366  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
5466  022370  104212                          ERROR   212             ;MSH B0 ERROR
5467  022372  104213                          ERROR   213             ;MSG A1 ERROR
5468  022374  104214                          ERROR   214             ;MSG B1 ERROR
5469  022376  004737  045534                  JSR     PC,SUBCLR
5470  022402  104024                          ERROR   24              ;CERR AFTER SCLR
5471
5472  022404  012737  000017  005314          MOV     #SEEK,HCS1
5473  022412  004737  043372                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
5474  022416  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
5475
5476  022420  013737  001412  005352          MOV     T50000,TEMP1    ;SETUP TIMEOUT
5477  022426  004737  044106                  JSR     PC,FATT2        ;FIND ATTN
5478  022432  104132                           ERROR   132             ;NO ATTN AFTER SEEK CMD
5479
5480  022434  032737  100000  005314          BIT     #CERR,HCS1
5481  022442  001401                          BEQ     65$
5482  022444  104210                          ERROR   210             ;CERR AFTER SEEK CMD
5483
5484  022446                          65$:
5485
5486  022446  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5487  022454  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5488  022460  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5489  022466  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5490  022474  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5491  022500  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5492  022506  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5493
5494  022514  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5495  022520  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5496  022522  104161                          ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
5497  022524  104162                          ERROR   162             ;MSH B0 ERROR
5498  022526  104163                          ERROR   163             ;MSG A1 ERROR
5499  022530  104164                          ERROR   164             ;MSG B1 ERROR
5500
5501
5502  022532  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
5503  022540  112737  000102  056576          MOVB    #'B,MSG19A
5504  022546  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
5505  022552  000401                          BR      1$
5506  022554  104103                          ERROR   103             ;PORT B AVAIL BEFORE TMO OR RELEASE
5507
5508  022556  032737  100000  005314  1$:     BIT     #CERR,HCS1
5509  022564  001001                          BNE     2$
5510  022566  104130                          ERROR   130             ;CERR NOT SET AFTER SEL DRIVE CMD
5511                                                                  ;& NO DRA
5512  022570  004737  043750          2$:     JSR     PC,TSTATN
5513  022574  000401                          BR      3$
```

```
5514  022576  104115                          ERROR   115             ;ATTN SET IN PORT B AFTER RLS FROM PORT A
5515
5516  022600  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
5517  022606  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5518  022614  012737  000000  005464          MOV     #0,UNITB
5519  022622  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5520  022630  112737  000101  056576          MOVB    #'A,MSG19A
5521  022636  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
5522  022644  012737  000001  005314          MOV     #SELDRV,HCS1
5523  022652  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5524  022656  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5525
5526  022660  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5527  022666  012737  000001  005464          MOV     #1,UNITB
5528  022674  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5529  022702  112737  000102  056576          MOVB    #'B,MSG19A
5530  022710  012737  000001  005314          MOV     #SELDRV,HCS1
5531  022716  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5532  022722  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5533
5534  022724  032737  000041  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
5535  022732  001001                          BNE     66$             ;BR IF YES
5536  022734  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
5537  022736                          66$:
5538  022736  004737  043750                  JSR     PC,TSTATN
5539  022742  104122                          ERROR   122             ;NO ATTN IN PORT B AFTER RLS FROM PORT A
5540
5541
5542  022744  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5543  022752  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5544  022756  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ,EXPECTED A1
5545  022764  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5546  022772  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5547  022776  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5548  023004  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5549
5550  023012  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5551  023016  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5552  023020  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
5553  023022  104212                          ERROR   212             ;MSH B0 ERROR
5554  023024  104213                          ERROR   213             ;MSG A1 ERROR
5555  023026  104214                          ERROR   214             ;MSG B1 ERROR
5556  023030  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5557  023036  012737  000001  005464          MOV     #1,UNITB
5558  023044  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5559  023052  112737  000102  056576          MOVB    #'B,MSG19A
5560  023060  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
5561  023066  012737  000001  005314          MOV     #SELDRV,HCS1
5562  023074  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5563  023100  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5564
5565  023102  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5566  023110  012737  000000  005464          MOV     #0,UNITB
5567  023116  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5568  023124  112737  000101  056576          MOVB    #'A,MSG19A
5569  023132  012737  000001  005314          MOV     #SELDRV,HCS1
```

```
5570  023140  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5571  023144  104117                  ERROR   117             ;NO RKY AFTER SEL DRV CMD
5572
5573  023146  032737  000040  005342  BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
5574  023154  001001                  BNE     67$             ;BR IF YES
5575  023156  104071                  ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
5576  023160                      67$:
5577  023160  004737  043750          JSR     PC,TSTATN
5578  023164  104123                  ERROR   123             ;ATTN CLEARED IN PORT A AFT RLS FROM PORT B
5579
5580
5581  023166  012737  050340  005404  MOV     #<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5582  023174  005037  005406          CLR     E.B0            ;EXPECTED MSG B0
5583  023200  012737  001720  005410  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5584  023206  012737  000001  005412  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5585  023214  005037  005414          CLR     E.A2            ;EXPECTED MSG A2
5586  023220  012737  000002  005416  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5587  023226  012737  000003  005422  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5588
5589  023234  004737  044274          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5590  023240  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5591  023242  104211                  ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
5592  023244  104212                  ERROR   212             ;MSH B0 ERROR
5593  023246  104213                  ERROR   213             ;MSG A1 ERROR
5594  023250  104214                  ERROR   214             ;MSG B1 ERROR
5595  023252  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5596  023260  012737  000000  005464  MOV     #0,UNITB
5597  023266  063765  005464  000010  ADD     UNITB,RKCS2(R5)
5598  023274  112737  000101  056576  MOVB    #'A,MSG19A
5599  023302  062765  000010  000010  ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
5600  023310  012737  000001  005314  MOV     #SELDRV,HCS1
5601  023316  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5602  023322  104117                  ERROR   117             ;NO RDY AFTER SEL DRV CMD
5603
5604  023324  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5605  023332  012737  000001  005464  MOV     #1,UNITB
5606  023340  063765  005464  000010  ADD     UNITB,RKCS2(R5)
5607  023346  112737  000102  056576  MOVB    #'B,MSG19A
5608  023354  012737  000001  005314  MOV     #SELDRV,HCS1
5609  023362  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5610  023366  104117                  ERROR   117             ;NO RKY AFTER SEL DRV CMD
5611
5612  023370  032737  000040  005342  BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
5613  023376  001001                  BNE     68$             ;BR IF YES
5614  023400  104071                  ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
5615  023402                      68$:
5616
5617  023402  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5618  023410  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5619  023416  063765  005464  000010  ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
5620  023424  012737  000005  005314  MOV     #CLEAR,HCS1
5621  023432  004737  043372          JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5622  023436  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5623  023440  004737  043750          JSR     PC,TSTATN       ;TEST FOR ATTN
5624  023444  000401                  BR      69$
5625  023446  104154                  ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
```

```
5626  023450                                69$:
5627
5628  023450  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5629  023456  005037  005406                  CLR     E.B0              ;EXPECTED MSG B0
5630  023462  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5631  023470  012737  000001  005412          MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
5632  023476  005037  005414                  CLR     E.A2              ;EXPECTED MSG A2
5633  023502  012737  000002  005416          MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
5634  023510  012737  000003  005422          MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
5635
5636  023516  004737  044274                  JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
5637  023522  000003                          .WORD   T.A2!T.B2!0               ;& MSGS SPECIFIED HERE
5638  023524  104033                          ERROR   33                ;MSG A0 ERROR AFTER DRV CLEAR CMD
5639  023526  104034                          ERROR   34                ;MSH B0 ERROR
5640  023530  104035                          ERROR   35                ;MSG A1 ERROR
5641  023532  104036                          ERROR   36                ;MSG B1 ERROR
5642
5643  023534  013737  005466  001366          MOV     TIMER,COUNT
5644  023542  004737  047204                  JSR     PC,TMO            ;DO 1.5 SEC TIMEOUT ON PORT B
5645  023546  004737  043750                  JSR     PC,TSTATN
5646  023552  000401                          BR      TST17             ;;GOTO NEXT TST
5647  023554  104144                          ERROR   144               ;MULT ATTN ON PORT B
5648
5649                                ;;*****************************************************************
5650                                ;*TEST 17       TEST RELEASE FROM PORT 'B' WITH PORT 'A' REQUESTING
5651                                ;*
5652                                ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
5653                                ;*
5654                                ;;*****************************************************************
5655  023556  000004                TST17:  SCOPE
5656  023560  012737  000001  001174          MOV     #1,$TIMES         ;;DO 1 ITERATION
5657  023566  012706  001100                  MOV     #STACK,SP
5658  023572  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5659  023600  012737  000000  005464          MOV     #0,UNITB
5660  023606  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5661  023614  112737  000101  056576          MOVB    #'A,MSG19A
5662  023622  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
5663  023630  012737  000001  005314          MOV     #SELDRV,HCS1
5664  023636  004737  043372                  JSR     PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5665  023642  104117                          ERROR   117               ;NO RDY AFTER SEL DRV CMD
5666
5667  023644  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5668  023652  012737  000001  005464          MOV     #1,UNITB
5669  023660  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5670  023666  112737  000102  056576          MOVB    #'B,MSG19A
5671  023674  012737  000001  005314          MOV     #SELDRV,HCS1
5672  023702  004737  043372                  JSR     PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5673  023706  104117                          ERROR   117               ;NO RKY AFTER SEL DRV CMD
5674
5675  023710  032737  000040  005342          BIT     #D.DRA,HMR2       ;SEE IF DRIVE AVAIL ON PORT B
5676  023716  001001                          BNE     64$               ;BR IF YES
5677  023720  104071                          ERROR   71                ;PORT B NOT AVAIL AFTER PORT A RLS
5678  023722                                64$:
5679
5680  023722  012737  010340  005404          MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5681  023730  005037  005406                  CLR     E.B0              ;EXPECTED MSG B0
```

```
5682  023734  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5683  023742  012737  000001  005412          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5684  023750  005037  005414                  CLR     E.A2                ;EXPECTED MSG A2
5685  023754  012737  000002  005416          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5686  023762  012737  000003  005422          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5687
5688  023770  004737  044274                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5689  023774  000000                          .WORD   0!0!0               ;& MSGS SPECIFIED HERE
5690  023776  104211                          ERROR   211                 ;MSG A0 ERROR AFTER RELEASE ISSUED
5691  024000  104212                          ERROR   212                 ;MSH B0 ERROR
5692  024002  104213                          ERROR   213                 ;MSG A1 ERROR
5693  024004  104214                          ERROR   214                 ;MSG B1 ERROR
5694  024006  004737  045534                  JSR     PC,SUBCLR
5695  024012  104024                          ERROR   24                  ;CERR AFTER SCLR
5696
5697  024014  012737  000017  005314          MOV     #SEEK,HCS1
5698  024022  004737  043372                  JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
5699  024026  104131                          ERROR   131                 ;NO RDY AFTER SEEK CMD
5700
5701  024030  013737  001412  005352          MOV     T50000,TEMP1        ;SETUP TIMEOUT
5702  024036  004737  044106                  JSR     PC,FATT2            ;FIND ATTN
5703  024042  104132                          ERROR   132                 ;NO ATTN AFTER SEEK CMD
5704
5705  024044  032737  100000  005314          BIT     #CERR,HCS1
5706  024052  001401                          BEQ     65$
5707  024054  104210                          ERROR   210                 ;CERR AFTER SEEK CMD
5708
5709  024056                          65$:
5710
5711  024056  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5712  024064  005037  005406                  CLR     E.B0                ;EXPECTED MSG B0
5713  024070  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5714  024076  012737  000001  005412          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5715  024104  005037  005414                  CLR     E.A2                ;EXPECTED MSG A2
5716  024110  012737  000002  005416          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5717  024116  012737  000003  005422          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5718
5719  024124  004737  044274                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5720  024130  000000                          .WORD   0!0!0               ;& MSGS SPECIFIED HERE
5721  024132  104161                          ERROR   161                 ;MSG A0 ERROR AFTER SEEK CMD
5722  024134  104162                          ERROR   162                 ;MSH B0 ERROR
5723  024136  104163                          ERROR   163                 ;MSG A1 ERROR
5724  024140  104164                          ERROR   164                 ;MSG B1 ERROR
5725
5726
5727  024142  012737  000000  005464          MOV     #0,UNITB            ;SETUP PORT A
5728  024150  112737  000101  056576          MOVB    #'A,MSG19A
5729  024156  004737  044172                  JSR     PC,DRAV             ;SEE IF DRIVE AVAIL
5730  024162  000401                          BR      1$
5731  024164  104103                          ERROR   103                 ;PORT A AVAIL BEFORE TMO OR RELEASE
5732
5733  024166  032737  100000  005314  1$:     BIT     #CERR,HCS1
5734  024174  001001                          BNE     2$
5735  024176  104130                          ERROR   130                 ;CERR NOT SET AFTER SEL DRIVE CMD
5736                                                                      ;& NO DRA
5737  024200  004737  043750          2$:     JSR     PC,TSTATN
```

```
5738  024204  000401                          BR      3$
5739  024206  104115                          ERROR   115             ;ATTN SET IN PORT A AFTER RLS FROM PORT B
5740
5741  024210  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
5742  024216  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5743  024224  012737  000001  005464          MOV     #1,UNITB
5744  024232  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5745  024240  112737  000102  056576          MOVB    #'B,MSG19A
5746  024246  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
5747  024254  012737  000001  005314          MOV     #SELDRV,HCS1
5748  024262  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5749  024266  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5750
5751  024270  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5752  024276  012737  000000  005464          MOV     #0,UNITB
5753  024304  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5754  024312  112737  000101  056576          MOVB    #'A,MSG19A
5755  024320  012737  000001  005314          MOV     #SELDRV,HCS1
5756  024326  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5757  024332  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5758
5759  024334  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
5760  024342  001001                          BNE     66$             ;BR IF YES
5761  024344  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
5762  024346                          66$:
5763  024346  004737  043750                  JSR     PC,TSTATN
5764  024352  104122                          ERROR   122             ;NO ATTN IN PORT A AFTER RLS FROM PORT B
5765
5766
5767  024354  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5768  024362  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5769  024366  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5770  024374  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5771  024402  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5772  024406  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5773  024414  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5774
5775  024422  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5776  024426  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5777  024430  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
5778  024432  104212                          ERROR   212             ;MSH B0 ERROR
5779  024434  104213                          ERROR   213             ;MSG A1 ERROR
5780  024436  104214                          ERROR   214             ;MSG B1 ERROR
5781  024440  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5782  024446  012737  000000  005464          MOV     #0,UNITB
5783  024454  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5784  024462  112737  000101  056576          MOVB    #'A,MSG19A
5785  024470  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
5786  024476  012737  000001  005314          MOV     #SELDRV,HCS1
5787  024504  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5788  024510  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5789
5790  024512  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5791  024520  012737  000001  005464          MOV     #1,UNITB
5792  024526  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5793  024534  112737  000102  056576          MOVB    #'B,MSG19A
```

```
5794  024542  012737  000001  005314          MOV     #SELDRV,HCS1
5795  024550  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5796  024554  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5797
5798  024556  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
5799  024564  001001                          BNE     67$             ;BR IF YES
5800  024566  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
5801  024570                          67$:
5802  024570  004737  043750                  JSR     PC,TSTATN
5803  024574  104123                          ERROR   123             ;ATTN CLEARED IN PORT B AFT RLS FROM PORT A
5804
5805
5806  024576  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5807  024604  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5808  024610  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5809  024616  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5810  024624  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5811  024630  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5812  024636  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5813
5814  024644  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5815  024650  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5816  024652  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
5817  024654  104212                          ERROR   212             ;MSH B0 ERROR
5818  024656  104213                          ERROR   213             ;MSG A1 ERROR
5819  024660  104214                          ERROR   214             ;MSG B1 ERROR
5820  024662  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5821  024670  012737  000001  005464          MOV     #1,UNITB
5822  024676  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5823  024704  112737  000102  056576          MOVB    #'B,MSG19A
5824  024712  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
5825  024720  012737  000001  005314          MOV     #SELDRV,HCS1
5826  024726  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5827  024732  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5828
5829  024734  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5830  024742  012737  000000  005464          MOV     #0,UNITB
5831  024750  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5832  024756  112737  000101  056576          MOVB    #'A,MSG19A
5833  024764  012737  000001  005314          MOV     #SELDRV,HCS1
5834  024772  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5835  024776  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5836
5837  025000  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
5838  025006  001001                          BNE     68$             ;BR IF YES
5839  025010  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B PLS
5840  025012                          68$:
5841
5842  025012  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5843  025020  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5844  025026  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
5845  025034  012737  000005  005314          MOV     #CLEAR,HCS1
5846  025042  004737  043372                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5847  025046  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5848  025050  004737  043750                  JSR     PC,TSTATN       ;TEST FOR ATTN
5849  025054  000401                          BR      69$
```

```
5850  025056  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5851  025060                          69$:
5852
5853  025060  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5854  025066  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5855  025072  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5856  025100  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5857  025106  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5858  025112  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5859  025120  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5860
5861  025126  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5862  025132  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5863  025134  104033                          ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
5864  025136  104034                          ERROR   34              ;MSH B0 ERROR
5865  025140  104035                          ERROR   35              ;MSG A1 ERROR
5866  025142  104036                          ERROR   36              ;MSG B1 ERROR
5867
5868  025144  013737  005466  001366          MOV     TIMER,COUNT
5869  025152  004737  047204                  JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT ON PORT A
5870  025156  004737  043750                  JSR     PC,TSTATN
5871  025162  000401                          BR      TST20           ;;GOTO NEXT TST
5872  025164  104144                          ERROR   144             ;MULT ATTN ON PORT A
5873
5874
5875                                  ;;**************************************************************
5876                                  ;*TEST 20       TEST RELEASE FROM REQUESTING PORT 'B' INHIBITS 'ATTN'
5877                                  ;*
5878                                  ;*      A.  PORT 'A' SEIZES THE DRIVE
5879                                  ;*      B.  PORT 'B' ATTEMPTS TO SEIZE THE DRIVE
5880                                  ;*      C.  PORT 'B' & PORT 'A' RELEASE THE DRIVE, IN THAT ORDER
5881                                  ;*      D.  THE PROGRAM VERIFIES THAT NEITHER PORT 'A' OR 'B' ATTENTION
5882                                  ;*          BITS SET
5883                                  ;*
5884                                  ;;**************************************************************
5885  025166  000004                  TST20:  SCOPE
5886  025170  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5887  025176  012706  001100                  MOV     #STACK,SP
5888  025202  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5889  025210  012737  000001  005464          MOV     #1,UNITB
5890  025216  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5891  025224  112737  000102  056576          MOVB    #'B,MSG19A
5892  025232  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
5893  025240  012737  000001  005314          MOV     #SELDRV,HCS1
5894  025246  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5895  025252  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5896
5897  025254  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5898  025262  012737  000000  005464          MOV     #0,UNITB
5899  025270  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5900  025276  112737  000101  056576          MOVB    #'A,MSG19A
5901  025304  012737  000001  005314          MOV     #SELDRV,HCS1
5902  025312  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5903  025316  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5904
5905  025320  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
```

```
5906  025326  001001                        BNE     64$             ;BR IF YES
5907  025330  104071                        ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
5908  025332                          64$:
5909
5910  025332  012737  010340  005404        MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5911  025340  005037  005406                CLR     E.B0            ;EXPECTED MSG B0
5912  025344  012737  001720  005410        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5913  025352  012737  000001  005412        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5914  025360  005037  005414                CLR     E.A2            ;EXPECTED MSG A2
5915  025364  012737  000002  005416        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5916  025372  012737  000003  005422        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5917
5918  025400  004737  044274                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5919  025404  000000                        .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5920  025406  104211                        ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
5921  025410  104212                        ERROR   212             ;MSH B0 ERROR
5922  025412  104213                        ERROR   213             ;MSG A1 ERROR
5923  025414  104214                        ERROR   214             ;MSG B1 ERROR
5924  025416  004737  045534                JSR     PC,SUBCLR
5925  025422  104024                        ERROR   24              ;CERR AFTER SCLR
5926
5927  025424  012737  000001  005464        MOV     #1,UNITB        ;SETUP PORT B
5928  025432  112737  000102  056576        MOVB    #'B,MSG19A
5929  025440  004737  044172                JSR     PC.DRAV         ;SEE IF DRIVE AVAIL
5930  025444  000401                        BR      1$
5931  025446  104103                        ERROR   103             ;PORT B AVAIL BEFORE TMO OR RELEASE
5932
5933  025450  032737  100000  005314  1$:   BIT     #CERR,HCS1
5934  025456  001001                        BNE     2$
5935  025460  104130                        ERROR   130             ;CERR NOT SET BY NO DRA
5936
5937  025462  004737  043750          2$:   JSR     PC,TSTATN
5938  025466  000401                        BR      3$
5939  025470  104115                        ERROR   115             ;ATTN SET IN PORT B AFTER RLS FROM PORT A
5940
5941  025472  012765  100000  000000  3$:   MOV     #CCLR,RKCS1(R5)
5942  025500  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5943  025506  012737  000001  005464        MOV     #1,UNITB
5944  025514  063765  005464  000010        ADD     UNITB,RKCS2(R5)
5945  025522  112737  000102  056576        MOVB    #'B,MSG19A
5946  025530  062765  000010  000010        ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
5947  025536  012737  000001  005314        MOV     #SELDRV,HCS1
5948  025544  004737  043372                JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5949  025550  104117                        ERROR   117             ;NO RDY AFTER SEL DRV CMD
5950
5951  025552  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5952  025560  012737  000000  005464        MOV     #0,UNITB
5953  025566  063765  005464  000010        ADD     UNITB,RKCS2(R5)
5954  025574  112737  000101  056576        MOVB    #'A,MSG19A
5955  025602  012737  000001  005314        MOV     #SELDRV,HCS1
5956  025610  004737  043372                JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5957  025614  104117                        ERROR   117             ;NO RKY AFTER SEL DRV CMD
5958
5959  025616  032737  000040  005342        BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
5960  025624  001001                        BNE     65$             ;BR IF YES
5961  025626  104071                        ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
```

```
5962  025630                                  65$:
5963  025630  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5964  025636  012737  000000  005464          MOV     #0,UNITB
5965  025644  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5966  025652  112737  000101  056576          MOVB    #'A,MSG19A
5967  025660  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
5968  025666  012737  000001  005314          MOV     #SELDRV,HCS1
5969  025674  004737  043372                  JSR     PC,DOCMD        ;DC SELDRV (STATUS) CMD & GET CONTR RDY
5970  025700  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
5971
5972  025702  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5973  025710  012737  000001  005464          MOV     #1,UNITB
5974  025716  063765  005464  000010          ADD     UNITB,RKCS2(R5)
5975  025724  112737  000102  056576          MOVB    #'B,MSG19A
5976  025732  012737  000001  005314          MOV     #SELDRV,HCS1
5977  025740  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5978  025744  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
5979
5980  025746  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
5981  025754  001001                          BNE     66$             ;BR IF YES
5982  025756  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
5983  025760                                  66$:
5984  025760  013737  005466  001366          MOV     TIMER,COUNT
5985  025766  004737  047204                  JSR     PC.TMO          ;DO 1.5 SEC TIMEOUT ON PORT B
5986  025772  004737  043750                  JSR     PC,TSTATN
5987  025776  000401                          BR      4$
5988  026000  104115                          ERROR   115             ;ATTN SET ON PORT B AFTER RLS FROM PORT B
5989
5990  026002                                  4$:
5991
5992  026002  012737  010340  005404          MOV     #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
5993  026010  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
5994  026014  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5995  026022  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5996  026030  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
5997  026034  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5998  026042  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5999
6000  026050  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6001  026054  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6002  026056  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
6003  026060  104212                          ERROR   212             ;MSH B0 ERROR
6004  026062  104213                          ERROR   213             ;MSG A1 ERROR
6005  026064  104214                          ERROR   214             ;MSG B1 ERROR
6006  026066  012737  000000  005464          MOV     #0,UNITB        ;SETUP PORT A
6007  026074  112737  000101  056576          MOVB    #'A,MSG19A
6008  026102  004737  043750                  JSR     PC,TSTATN
6009  026106  000401                          BR      TST21           ;;GOTO NEXT TST
6010  026110  104115                          ERROR   115             ;ATTN SET ON PORT A, AFTER RLS FROM PORT A
6011
6012                                  ;;**********************************************************************
6013                                  ;*TEST 21        TEST RELEASE FROM REQUESTING PORT 'A' INHIBITS 'ATTN'
6014                                  ;*
6015                                  ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6016                                  ;*
6017                                  ;;**********************************************************************
```

```
6018  026112  000004                          TST21:  SCOPE
6019  026114  012737  000001  001174                  MOV     #1,$TIMES       ;;DO 1 ITERATION
6020  026122  012706  001100                          MOV     #STACK,SP
6021  026126  013765  001222  000010                  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6022  026134  012737  000000  005464                  MOV     #0,UNITB
6023  026142  063765  005464  000010                  ADD     UNITB,RKCS2(R5)
6024  026150  112737  000101  056576                  MOVB    #'A,MSG19A
6025  026156  062765  000010  000010                  ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6026  026164  012737  000001  005314                  MOV     #SELDRV,HCS1
6027  026172  004737  043372                          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6028  026176  104117                                  ERROR   117             ;NO RDY AFTER SEL DRV CMD
6029
6030  026200  013765  001222  000010                  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6031  026206  012737  000001  005464                  MOV     #1,UNITB
6032  026214  063765  005464  000010                  ADD     UNITB,RKCS2(R5)
6033  026222  112737  000102  056576                  MOVB    #'B,MSG19A
6034  026230  012737  000001  005314                  MOV     #SELDRV,HCS1
6035  026236  004737  043372                          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6036  026242  104117                                  ERROR   117             ;NO RKY AFTER SEL DRV CMD
6037
6038  026244  032737  000040  005342                  BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6039  026252  001001                                  BNE     64$             ;BR IF YES
6040  026254  104071                                  ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6041  026256                                  64$:
6042
6043  026256  012737  010340  005404                  MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6044  026264  005037  005406                          CLR     E.B0            ;EXPECTED MSG B0
6045  026270  012737  001720  005410                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6046  026276  012737  000001  005412                  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6047  026304  005037  005414                          CLR     E.A2            ;EXPECTED MSG A2
6048  026310  012737  000002  005416                  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6049  026316  012737  000003  005422                  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6050
6051  026324  004737  044274                          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6052  026330  000000                                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6053  026332  104211                                  ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
6054  026334  104212                                  ERROR   212             ;MSH B0 ERROR
6055  026336  104213                                  ERROR   213             ;MSG A1 ERROR
6056  026340  104214                                  ERROR   214             ;MSG B1 ERROR
6057  026342  004737  045534                          JSR     PC,SUBCLR
6058  026346  104024                                  ERROR   24              ;CERR AFTER SCLR
6059
6060  026350  012737  000000  005464                  MOV     #0,UNITB        ;SETUP PORT A
6061  026356  112737  000101  056576                  MOVB    #'A,MSG19A
6062  026364  004737  044172                          JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
6063  026370  000401                                  BR      1$
6064  026372  104103                                  ERROR   103             ;PORT A AVAIL BEFORE TMO OR RELEASE
6065
6066  026374  032737  100000  005314  1$:             BIT     #CERR,HCS1
6067  026402  001001                                  BNE     2$
6068  026404  104130                                  ERROR   130             ;CERR NOT SET BY NO DRA
6069
6070  026406  004737  043750          2$:             JSR     PC,TSTATN
6071  026412  000401                                  BR      3$
6072  026414  104115                                  ERROR   115             ;ATTN SET IN PORT A AFTER RLS FROM PORT B
6073
```

```
6074  026416  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
6075  026424  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6076  026432  012737  000000  005464          MOV     #0,UNITB
6077  026440  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6078  026446  112737  000101  056576          MOVB    #'A,MSG19A
6079  026454  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6080  026462  012737  000001  005314          MOV     #SELDRV,HCS1
6081  026470  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6082  026474  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6083
6084  026476  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6085  026504  012737  000001  005464          MOV     #1,UNITB
6086  026512  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6087  026520  112737  000102  056576          MOVB    #'B,MSG19A
6088  026526  012737  000001  005314          MOV     #SELDRV,HCS1
6089  026534  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6090  026540  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6091
6092  026542  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6093  026550  001001                          BNE     65$             ;BR IF YES
6094  026552  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6095  026554                          65$:
6096  026554  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6097  026562  012737  000001  005464          MOV     #1,UNITB
6098  026570  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6099  026576  112737  000102  056576          MOVB    #'B,MSG19A
6100  026604  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
6101  026612  012737  000001  005314          MOV     #SELDRV,HCS1
6102  026620  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6103  026624  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6104
6105  026626  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6106  026634  012737  000000  005464          MOV     #0,UNITB
6107  026642  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6108  026650  112737  000101  056576          MOVB    #'A,MSG19A
6109  026656  012737  000001  005314          MOV     #SELDRV,HCS1
6110  026664  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6111  026670  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6112
6113  026672  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
6114  026700  001001                          BNE     66$             ;BR IF YES
6115  026702  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
6116  026704                          66$:
6117  026704  013737  005466  001366          MOV     TIMER,COUNT
6118  026712  004737  047204                  JSR     PC,TMO          ;DO 1.5 SEC TIMEOUT ON PORT A
6119  026716  004737  043750                  JSR     PC,TSTATN
6120  026722  000401                          BR      4$
6121  026724  104115                          ERROR   115             ;ATTN SET ON PORT A AFTER RLS FROM PORT A
6122
6123  026726                          4$:
6124
6125  026726  012737  010340  005404          MOV     #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6126  026734  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
6127  026740  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6128  026746  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6129  026754  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
```

```
6130  026760  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6131  026766  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6132
6133  026774  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6134  027000  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6135  027002  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
6136  027004  104212                          ERROR   212             ;MSH B0 ERROR
6137  027006  104213                          ERROR   213             ;MSG A1 ERROR
6138  027010  104214                          ERROR   214             ;MSG B1 ERROR
6139  027012  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
6140  027020  112737  000102  056576          MOVB    #'B,MSG19A
6141  027026  004737  043750                  JSR     PC,TSTATN
6142  027032  000401                          BR      TST22           ;;GOTO NEXT TST
6143  027034  104115                          ERROR   115             ;ATTN SET ON PORT B, AFTER RLS FROM PORT B
6144
6145                                  ;;**************************************************************
6146                                  ;*TEST 22       TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
6147                                  ;*
6148                                  ;*      VERIFY THAT A RELEASE ISSUED BY ONE PORT IS NOT RECOGNIZED IF
6149                                  ;*      THE DRIVE IS SEIZED BY THE OTHER PORT
6150                                  ;*
6151                                  ;*      A.   SEIZE THE DRIVE THRU PORT 'A'.
6152                                  ;*
6153                                  ;*      B.   ISSUE A RELEASE THRU PORT 'B' & VERIFY DRIVE STILL SEIZED
6154                                  ;*           BY PORT 'A'.
6155                                  ;*
6156                                  ;;**************************************************************
6157  027036  000004                  TST22:  SCOPE
6158  027040  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6159  027046  012706  001100                  MOV     #STACK,SP
6160  027052  012737  000000  005464          MOV     #0,UNITB        ;SETUP PORT A
6161  027060  112737  000101  056576          MOVB    #'A,MSG19A
6162  027066  013737  005466  001366          MOV     TIMER,COUNT
6163  027074  004737  047204                  JSR     PC,TMO          ;DO TIMEOUT
6164
6165  027100  004737  045534                  JSR     PC,SUBCLR
6166  027104  104024                          ERROR   24              ;CERR AFTER SCLR
6167
6168
6169  027106  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
6170  027112  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TMO
6171  027114  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6172  027122  012737  000001  005464          MOV     #1,UNITB
6173  027130  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6174  027136  112737  000102  056576          MOVB    #'B,MSG19A
6175  027144  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
6176  027152  012737  000001  005314          MOV     #SELDRV,HCS1
6177  027160  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6178  027164  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6179
6180  027166  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6181  027174  012737  000000  005464          MOV     #0,UNITB
6182  027202  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6183  027210  112737  000101  056576          MOVB    #'A,MSG19A
6184  027216  012737  000001  005314          MOV     #SELDRV,HCS1
6185  027224  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
```

```
 6186  027230  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
 6187
 6188  027232  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
 6189  027240  001001                          BNE     64$             ;BR IF YES
 6190  027242  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
 6191  027244                          64$:
 6192
 6193  027244  012737  010340  005404          MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 6194  027252  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
 6195  027256  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 6196  027264  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 6197  027272  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
 6198  027276  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 6199  027304  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 6200
 6201  027312  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 6202  027316  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 6203  027320  104211                          ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
 6204  027322  104212                          ERROR   212             ;MSH B0 ERROR
 6205  027324  104213                          ERROR   213             ;MSG A1 ERROR
 6206  027326  104214                          ERROR   214             ;MSG B1 ERROR
 6207
 6208                          ;;*****************************************************************
 6209                          ;*TEST 23       TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
 6210                          ;*
 6211                          ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
 6212                          ;*
 6213                          ;;*****************************************************************
 6214  027330  000004          TST23:  SCOPE
 6215  027332  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 6216  027340  012706  001100                  MOV     #STACK,SP
 6217  027344  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
 6218  027352  112737  000102  056576          MOVB    #'B,MSG19A
 6219  027360  013737  005466  001366          MOV     TIMER,COUNT
 6220  027366  004737  047204                  JSR     PC,TMO          ;DO TIMEOUT
 6221
 6222  027372  004737  045534                  JSR     PC,SUBCLR
 6223  027376  104024                          ERROR   24              ;CERR AFTER SCLR
 6224
 6225
 6226  027400  004737  044172                  JSR     PC,DRAV         ;SEE IF DRIVE AVAIL
 6227  027404  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TMO
 6228  027406  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
 6229  027414  012737  000000  005464          MOV     #0,UNITB
 6230  027422  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6231  027430  112737  000101  056576          MOVB    #'A,MSG19A
 6232  027436  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
 6233  027444  012737  000001  005314          MOV     #SELDRV,HCS1
 6234  027452  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 6235  027456  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
 6236
 6237  027460  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
 6238  027466  012737  000001  005464          MOV     #1,UNITB
 6239  027474  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6240  027502  112737  000102  056576          MOVB    #'B,MSG19A
 6241  027510  012737  000001  005314          MOV     #SELDRV,HCS1
```

```
6242  027516  004737  043372              JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6243  027522  104117                      ERROR   117             ;NO RKY AFTER SEL DRV CMD
6244
6245  027524  032737  000040  005342      BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6246  027532  001001                      BNE     64$             ;BR IF YES
6247  027534  104071                      ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6248  027536                          64$:
6249
6250  027536  012737  010340  005404      MOV     #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6251  027544  005037  005406              CLR     E.B0            ;EXPECTED MSG B0
6252  027550  012737  001720  005410      MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6253  027556  012737  000001  005412      MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6254  027564  005037  005414              CLR     E.A2            ;EXPECTED MSG A2
6255  027570  012737  000002  005416      MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6256  027576  012737  000003  005422      MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6257
6258  027604  004737  044274              JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6259  027610  000000                      .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6260  027612  104211                      ERROR   211             ;MSG A0 ERROR AFTER RELEASE ISSUED
6261  027614  104212                      ERROR   212             ;MSH B0 ERROR
6262  027616  104213                      ERROR   213             ;MSG A1 ERROR
6263  027620  104214                      ERROR   214             ;MSG B1 ERROR
6264
```

```
6265                                    ;;**********************************************************************
6266                                    ;*TEST 24       TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE: PORT A
6267                                    ;*
6268                                    ;*      A.   ISSUE A SEEK COMMAND TO CYL 10 FROM PORT 'A' & AN
6269                                    ;*           IMMEDIATE RELEASE TO PORT 'A'
6270                                    ;*
6271                                    ;*      B.   VERIFY THE DRIVE IS AVAILABLE TO PORT 'B' & PORT B SEES ATTN
6272                                    ;*
6273                                    ;*      C.   VERIFY PORT A DOES NOT RAISE ATTN WHEN SEEK COMPLETED.
6274
6275                                    ,*
6276                                    ;;**********************************************************************
6277  027622  000004                    TST24:  SCOPE
6278  027624  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
6279  027632  012706  001100                    MOV     #STACK,SP
6280  027636  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6281  027644  012737  000001  005464            MOV     #1,UNITB
6282  027652  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6283  027660  112737  000102  056576            MOVB    #'B,MSG19A
6284  027666  062765  000010  000010            ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
6285  027674  012737  000001  005314            MOV     #SELDRV,HCS1
6286  027702  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6287  027706  104117                            ERROR   117             ;NO RDY AFTER SEL DRV CMD
6288
6289  027710  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6290  027716  012737  000000  005464            MOV     #0,UNITB
6291  027724  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6292  027732  112737  000101  056576            MOVB    #'A,MSG19A
6293  027740  012737  000001  005314            MOV     #SELDRV,HCS1
6294  027746  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6295  027752  104117                            ERROR   117             ;NO RKY AFTER SEL DRV CMD
6296
6297  027754  032737  000040  005342            BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
6298  027762  001001                            BNE     64$             ;BR IF YES
6299  027764  104071                            ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
6300  027766                          64$:
6301  027766  004737  045534                    JSR     PC,SUBCLR
6302  027772  104024                            ERROR   24              ;CERR AFTER SCLR
6303
6304  027774  012765  000012  000020            MOV     #10.,RKDC(R5)
6305  030002  012737  000017  005314            MOV     #SEEK,HCS1
6306  030010  004737  043372                    JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
6307  030014  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD
6308  030016  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6309  030024  012737  000000  005464            MOV     #0,UNITB
6310  030032  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6311  030040  112737  000101  056576            MOVB    #'A,MSG19A
6312  030046  062765  000010  000010            ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6313  030054  012737  000001  005314            MOV     #SELDRV,HCS1
6314  030062  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6315  030066  104117                            ERROR   117             ;NO RDY AFTER SEL DRV CMD
6316
6317  030070  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6318  030076  012737  000001  005464            MOV     #1,UNITB
6319  030104  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6320  030112  112737  000102  056576            MOVB    #'B,MSG19A
```

```
6321  030120  012737  000001  005314          MOV     #SELDRV,HCS1
6322  030126  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6323  030132  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6324
6325  030134  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6326  030142  001001                          BNE     65$             ;BR IF YES
6327  030144  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6328  030146                          65$:
6329  030146  013737  001412  005352          MOV     T50000,TEMP1
6330  030154  004737  044106                  JSR     PC,FATT2
6331  030160  104152                          ERROR   152             ;NO ATTN ON PORT B AFTER SEEK & RLS FROM PORT A
6332
6333
6334  030162  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6335  030170  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
6336  030174  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6337  030202  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6338  030210  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
6339  030214  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6340  030222  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6341
6342  030230  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6343  030234  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6344  030236  104145                          ERROR   145             ;MSG A0 ERROR AFTER SEEK & RLS FROM PORT A
6345  030240  104146                          ERROR   146             ;MSH B0 ERROR
6346  030242  104147                          ERROR   147             ;MSG A1 ERROR
6347  030244  104150                          ERROR   150             ;MSG B1 ERROR
6348  030246  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6349  030254  012737  000001  005464          MOV     #1,UNITB
6350  030262  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6351  030270  112737  000102  056576          MOVB    #'B,MSG19A
6352  030276  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
6353  030304  012737  000001  005314          MOV     #SELDRV,HCS1
6354  030312  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6355  030316  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6356
6357  030320  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6358  030326  012737  000000  005464          MOV     #0,UNITB
6359  030334  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6360  030342  112737  000101  056576          MOVB    #'A,MSG19A
6361  030350  012737  000001  005314          MOV     #SELDRV,HCS1
6362  030356  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6363  030362  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6364
6365  030364  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
6366  030372  001001                          BNE     66$             ;BR IF YES
6367  030374  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
6368  030376                          66$:
6369  030376  004737  043750                  JSR     PC,TSTATN
6370  030402  000401                          BR      1$
6371  030404  104160                          ERROR   160             ;ATTN ON PORT A AFTER SEEK & RLS FROM PORT A
6372
6373  030406                          1$:
6374
6375  030406  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6376  030414  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
```

```
6377  030420  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6378  030426  012737  000001  005412          MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
6379  030434  005037  005414                  CLR     E.A2              ;EXPECTED MSG A2
6380  030440  012737  000002  005416          MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
6381  030446  012737  000003  005422          MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
6382
6383  030454  004737  044274                  JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
6384  030460  000000                          .WORD   0!0!0             ;& MSGS SPECIFIED HERE
6385  030462  104145                          ERROR   145               ;MSG A0 ERROR AFTER SEEK & RLS FROM PORT A
6386  030464  104146                          ERROR   146               ;MSH B0 ERROR
6387  030466  104147                          ERROR   147               ;MSG A1 ERROR
6388  030470  104150                          ERROR   150               ;MSG B1 ERROR
6389
6390                                  ;;*****************************************************************
6391                                  ;*TEST 25       TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE: PORT B
6392                                  ;*
6393                                  ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B',
6394                                  ;*      BUT THE SEEK IS TO CYLINDER 0
6395                                  ;*
6396                                  ;;*****************************************************************
6397  030472  000004                  TST25:  SCOPE
6398  030474  012737  000001  001174          MOV     #1,$TIMES         ;;DO 1 ITERATION
6399  030502  012706  001100                  MOV     #STACK,SP
6400  030506  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6401  030514  012737  000000  005464          MOV     #0,UNITB
6402  030522  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6403  030530  112737  000101  056576          MOVB    #'A,MSG19A
6404  030536  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6405  030544  012737  000001  005314          MOV     #SELDRV,HCS1
6406  030552  004737  043372                  JSR     PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6407  030556  104117                          ERROR   117               ;NO RDY AFTER SEL DRV CMD
6408
6409  030560  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6410  030566  012737  000001  005464          MOV     #1,UNITB
6411  030574  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6412  030602  112737  000102  056576          MOVB    #'B,MSG19A
6413  030610  012737  000001  005314          MOV     #SELDRV,HCS1
6414  030616  004737  043372                  JSR     PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6415  030622  104117                          ERROR   117               ;NO RKY AFTER SEL DRV CMD
6416
6417  030624  032737  000040  005342          BIT     #D.DRA,HMR2       ;SEE IF DRIVE AVAIL ON PORT B
6418  030632  001001                          BNE     64$               ;BR IF YES
6419  030634  104071                          ERROR   71                ;PORT B NOT AVAIL AFTER PORT A RLS
6420  030636                          64$:
6421  030636  004737  045534                  JSR     PC,SUBCLR
6422  030642  104024                          ERROR   24                ;CERR AFTER SCLR
6423
6424  030644  012765  000000  000020          MOV     #0,RKDC(R5)
6425  030652  012737  000017  005314          MOV     #SEEK,HCS1
6426  030660  004737  043372                  JSR     PC,DOCMD          ;DO SEEK CMD & GET CONTR READY
6427  030664  104131                          ERROR   131               ;NO RDY AFTER SEEK CMD
6428  030666  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6429  030674  012737  000001  005464          MOV     #1,UNITB
6430  030702  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6431  030710  112737  000102  056576          MOVB    #'B,MSG19A
6432  030716  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
```

```
6433  030724  012737  000001  005314          MOV     #SELDRV,HCS1
6434  030732  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6435  030736  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6436
6437  030740  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6438  030746  012737  000000  005464          MOV     #0,UNITB
6439  030754  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6440  030762  112737  000101  056576          MOVB    #'A,MSG19A
6441  030770  012737  000001  005314          MOV     #SELDRV,HCS1
6442  030776  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6443  031002  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6444
6445  031004  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
6446  031012  001001                          BNE     65$             ;BR IF YES
6447  031014  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
6448  031016                          65$:
6449  031016  013737  001412  005352          MOV     T50000,TEMP1
6450  031024  004737  044106                  JSR     PC,FATT2
6451  031030  104152                          ERROR   152             ;NO ATTN ON PORT A AFTER SEEK & RLS FROM PORT B
6452
6453
6454  031032  012737  050340  005404          MOV     #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6455  031040  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
6456  031044  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6457  031052  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6458  031060  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
6459  031064  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6460  031072  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6461
6462  031100  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6463  031104  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6464  031106  104145                          ERROR   145             ;MSG A0 ERROR AFTER SEEK & RLS FROM PORT B
6465  031110  104146                          ERROR   146             ;MSH B0 ERROR
6466  031112  104147                          ERROR   147             ;MSG A1 ERROR
6467  031114  104150                          ERROR   150             ;MSG B1 ERROR
6468  031116  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6469  031124  012737  000000  005464          MOV     #0,UNITB
6470  031132  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6471  031140  112737  000101  056576          MOVB    #'A,MSG19A
6472  031146  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6473  031154  012737  000001  005314          MOV     #SELDRV,HCS1
6474  031162  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6475  031166  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6476
6477  031170  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6478  031176  012737  000001  005464          MOV     #1,UNITB
6479  031204  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6480  031212  112737  000102  056576          MOVB    #'B,MSG19A
6481  031220  012737  000001  005314          MOV     #SELDRV,HCS1
6482  031226  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6483  031232  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6484
6485  031234  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6486  031242  001001                          BNE     66$             ;BR IF YES
6487  031244  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6488  031246                          66$:
```

```
6489  031246  004737  043750                  JSR     PC,TSTATN
6490  031252  000401                          BR      1$
6491  031254  104160                          ERROR   160             ;ATTN ON PORT B AFTER SEEK & RLS FROM PORT B
6492
6493  031256                          1$:
6494
6495  031256  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6496  031264  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
6497  031270  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6498  031276  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6499  031304  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
6500  031310  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6501  031316  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6502
6503  031324  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6504  031330  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6505  031332  104145                          ERROR   145             ;MSG A0 ERROR AFTER SEEK & RLS FROM PORT B
6506  031334  104146                          ERROR   146             ;MSH B0 ERROR
6507  031336  104147                          ERROR   147             ;MSG A1 ERROR
6508  031340  104150                          ERROR   150             ;MSG B1 ERROR
6509
6510
6511                          ;;*****************************************************************
6512                          ;*TEST 26       TEST TIMEOUT RETRIGGER THRU PORT 'A'
6513                          ;*
6514                          ;*      VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
6515                          ;*
6516                          ;*      A.  PORT 'A' SEIZES THE DRIVE
6517                          ;*
6518                          ;*      B.  THE PROGRAM WAITS 500MS & RE-SEIZES THE DRIVE THRU PORT 'A'
6519                          ;*
6520                          ;*      C.  PORT 'B' ATTEMPTS TO SEIZE THE DRIVE & THE PROGRAM
6521                          ;*          VERIFIES THAT FULL TIMEOUT TOOK PLACE FROM STEP B ABOVE.
6522                          ;*
6523                          ;;*****************************************************************
6524  031342  000004          TST26:  SCOPE
6525  031344  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6526  031352  012706  001100                  MOV     #STACK,SP
6527  031356  004737  045534                  JSR     PC,SUBCLR
6528  031362  104024                          ERROR   24              ;CERR AFTER SCLR
6529  031364  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6530  031372  012737  000001  005464          MOV     #1,UNITB
6531  031400  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6532  031406  112737  000102  056576          MOVB    #'B,MSG19A
6533  031414  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
6534  031422  012737  000001  005314          MOV     #SELDRV,HCS1
6535  031430  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6536  031434  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
6537
6538  031436  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6539  031444  012737  000000  005464          MOV     #0,UNITB
6540  031452  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6541  031460  112737  000101  056576          MOVB    #'A,MSG19A
6542  031466  012737  000001  005314          MOV     #SELDRV,HCS1
6543  031474  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6544  031500  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
```

```
6545
6546  031502  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
6547  031510  001001                          BNE     64$             ;BR IF YES
6548  031512  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
6549  031514                          64$:
6550  031514  004737  045534                  JSR     PC,SUBCLR
6551  031520  104024                          ERROR   24              ;CERR AFTER SCLR
6552
6553  031522  012737  000036  001366          MOV     #30.,COUNT
6554  031530  004737  047204                  JSR     PC,TMO          ;DO 500MS TIMEOUT
6555  031534  004737  044172                  JSR     PC,DRAV         ;RE-SEIZE DRIVE THRU PORT A
6556  031540  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TIMEOUT
6557
6558  031542  012737  010340  005404          MOV     #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6559  031550  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
6560  031554  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6561  031562  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6562  031570  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
6563  031574  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6564  031602  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6565
6566  031610  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6567  031614  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6568  031616  104165                          ERROR   165             ;MSG A0 ERROR AFTER TIMEOUT
6569  031620  104166                          ERROR   166             ;MSH B0 ERROR
6570  031622  104167                          ERROR   167             ;MSG A1 ERROR
6571  031624  104170                          ERROR   170             ;MSG B1 ERROR
6572  031626  012737  177777  001366          MOV     #-1,COUNT
6573  031634  004737  047114                  JSR     PC,CLKON        ;TURN ON CLOCK
6574  031640  012737  000001  005464          MOV     #1,UNITB        ;SETUP PORT B
6575  031646  004737  044172                  JSR     PC,DRAV         ;SEE IF DRV AVAIL
6576  031652  000401                          BR      1$              ;BR IF NO
6577  031654  104103                          ERROR   103             ;PORT A AVAIL BEFORE TMO OR RLS
6578  031656  112737  000102  056576  1$:     MOVB    #'B,MSG19A
6579  031664  013704  001222                  MOV     $UNIT,R4
6580  031670  063704  005464                  ADD     UNITB,R4
6581  031674  004737  044214                  JSR     PC,FATT3
6582  031700  104110                          ERROR   110             ;NO ATTN ON PORT B TO ALLOW SEIZE
6583
6584  031702  004737  047162                  JSR     PC,CLKOF        ;TURN CLOCK OFF
6585  031706  005137  001366                  COM     COUNT           ;GET ACTUAL COUNT OF TIMEOUT
6586  031712  023727  001366  000043          CMP     COUNT,#35.      ;COMPARE COUNT AGAINST APPROX 1 SEC
6587  031720  002001                          BGE     TST27           ;;GO TO NEXT TEST
6588  031722  104153                          ERROR   153             ;TIMEOUT DID NOT RE-TRIGGER
6589
6590                          ;,*****************************************************************
6591                          ;*TEST 27        TEST TIMEOUT RETRIGGER THRU PORT 'B'
6592                          ;*
6593                          ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6594                          ;*
6595                          ;;*****************************************************************
6596  031724  000004          TST27:  SCOPE
6597  031726  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6598  031734  012706  001100                  MOV     #STACK,SP
6599  031740  004737  045534                  JSR     PC,SUBCLR
6600  031744  104024                          ERROR   24              ;CERR AFTER SCLR
```

```
6601  031746  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6602  031754  012737  000000  005464          MOV     #0,UNITB
6603  031762  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6604  031770  112737  000101  056576          MOVB    #'A,MSG19A
6605  031776  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6606  032004  012737  000001  005314          MOV     #SELDRV,HCS1
6607  032012  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6608  032016  104117                          ERROR   117             ;NC RDY AFTER SEL DRV CMD
6609
6610  032020  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6611  032026  012737  000001  005464          MOV     #1,UNITB
6612  032034  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6613  032042  112737  000102  056576          MOVB    #'B,MSG19A
6614  032050  012737  000001  005314          MOV     #SELDRV,HCS1
6615  032056  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6616  032062  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
6617
6618  032064  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6619  032072  001001                          BNE     64$             ;BR IF YES
6620  032074  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6621  032076                          64$:
6622  032076  004737  045534                  JSR     PC,SUBCLR
6623  032102  104024                          ERROR   24              ;CERR AFTER SCLR
6624
6625  032104  012737  000036  001366          MOV     #30.,COUNT
6626  032112  004737  047204                  JSR     PC,TMO          ;DO 500MS TIMEOUT
6627  032116  004737  044172                  JSR     PC,DRAV         ;RE-SEIZE DRIVE THRU PORT B
6628  032122  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TIMEOUT
6629
6630  032124  012737  010340  005404          MOV     #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
6631  032132  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
6632  032136  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
6633  032144  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
6634  032152  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
6635  032156  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
6636  032164  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
6637
6638  032172  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
6639  032176  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
6640  032200  104165                          ERROR   165             ;MSG A0 ERROR AFTER TIMEOUT
6641  032202  104166                          ERROR   166             ;MSH B0 ERROR
6642  032204  104167                          ERROR   167             ;MSG A1 ERROR
6643  032206  104170                          ERROR   170             ;MSG B1 ERROR
6644  032210  012737  177777  001366          MOV     #-1,COUNT
6645  032216  004737  047114                  JSR     PC,CLKON        ;TURN ON CLOCK
6646  032222  012737  000000  005464          MOV     #0,UNITB        ;SETUP PORT A
6647  032230  004737  044172                  JSR     PC,DRAV         ;SEE IF DRV AVAIL
6648  032234  000401                          BR      1$              ;BR IF NO
6649  032236  104103                          ERROR   103             ;PORT B AVAIL BEFORE TMO OR RLS
6650  032240  112737  000101  056576  1$:     MOVB    #'A,MSG19A
6651  032246  013704  001222                  MOV     $UNIT,R4
6652  032252  063704  005464                  ADD     UNITB,R4
6653  032256  004737  044214                  JSR     PC,FATT3
6654  032262  104110                          ERROR   110             ;NO ATTN ON PORT A TO ALLOW SEIZE
6655
6656  032264  004737  047162                  JSR     PC,CLKOF        ;TURN CLOCK OFF
```

```
  6657  032270  005137  001366                COM     COUNT           ;GET ACTUAL COUNT OF TIMEOUT
  6658  032274  023727  001366  000043        CMP     COUNT,#35.      ;COMPARE COUNT AGAINST APPROX 1 SEC
  6659  032302  002001                        BGE     TST30           ;;GO TO NEXT TEST
  6660  032304  104153                        ERROR   153             ;TIMEOUT DID NOT RE-TRIGGER
  6661
```

```
                                        ;;**********************************************************************
                                        ;*TEST 30       TEST PORT 'A' TIMER INHIBIT
                                        ;*
                                        ;*      A.   PORT 'A' SEIZES THE DRIVE
                                        ;*      B.   PORT 'B' ATTEMPTS TO SEIZE THE DRIVE
                                        ;*      C.   PORT 'A' RELEASES THE DRIVE
                                        ;*      D.   PORT 'A' ATTEMPTS TO GET THE DRIVE BACK.
                                        ;*
                                        ;*      THE PROGRAM VERIFIES THAT PORT 'A' CANNOT ACCESS
                                        ;*      THE DRIVE FOR APPROX 1 SEC
                                        ,*

                                        ;;**********************************************************************
032306  000004                    TST30:  SCOPE
032310  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
032316  012706  001100                    MOV     #STACK,SP
032322  004737  045534                    JSR     PC,SUBCLR
032326  104024                            ERROR   24              ;CERR AFTER SCLR
032330  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
032336  012737  000001  005464            MOV     #1,UNITB
032344  063765  005464  000010            ADD     UNITB,RKCS2(R5)
032352  112737  000102  056576            MOVB    #'B,MSG19A
032360  062765  000010  000010            ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
032366  012737  000001  005314            MOV     #SELDRV,HCS1
032374  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
032400  104117                            ERROR   117             ;NO RDY AFTER SEL DRV CMD

032402  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
032410  012737  000000  005464            MOV     #0,UNITB
032416  063765  005464  000010            ADD     UNITB,RKCS2(R5)
032424  112737  000101  056576            MOVB    #'A,MSG19A
032432  012737  000001  005314            MOV     #SELDRV,HCS1
032440  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
032444  104117                            ERROR   117             ;NO RKY AFTER SEL DRV CMD

032446  032737  000040  005342            BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
032454  001001                            BNE     64$             ;BR IF YES
032456  104071                            ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
032460                          64$:
032460  004737  045534                    JSR     PC,SUBCLR
032464  104024                            ERROR   24              ;CERR AFTER SCLR

032466  012737  000001  005464            MOV     #1,UNITB        ;SETUP FOR PORT B
032474  112737  000102  056576            MOVB    #'B,MSG19A
032502  004737  044172                    JSR     PC,DRAV         ;PORT B TRIES TO SEIZE THE DRIVE
032506  000401                            BR      1$              ;BR IF NOT AVAIL
032510  104103                            ERROR   103             ;PORT B AVAIL BEFORE TMO OR RELEASE

032512  012765  100000  000000    1$:     MOV     #CCLR,RKCS1(R5)
032520  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
032526  012737  000000  005464            MOV     #0,UNITB
032534  063765  005464  000010            ADD     UNITB,RKCS2(R5)
032542  112737  000101  056576            MOVB    #'A,MSG19A
032550  062765  000010  000010            ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
032556  012737  000001  005314            MOV     #SELDRV,HCS1
032564  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
```

```
 6718  032570  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
 6719
 6720  032572  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
 6721  032600  012737  000001  005464          MOV     #1,UNITB
 6722  032606  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6723  032614  112737  000102  056576          MOVB    #'B,MSG19A
 6724  032622  012737  000001  005314          MOV     #SELDRV,HCS1
 6725  032630  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 6726  032634  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
 6727
 6728  032636  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
 6729  032644  001001                          BNE     65$             ;BR IF YES
 6730  032646  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
 6731  032650                          65$:
 6732  032650  012737  000000  005464          MOV     #0,UNITB        ;SETUP FOR PORT A
 6733  032656  112737  000101  056576          MOVB    #'A,MSG19A
 6734  032664  004737  044172                  JSR     PC,DRAV
 6735  032670  000401                          BR      2$
 6736  032672  104103                          ERROR   103             ;PORT A AVAIL BEFORE TMO OR RELEASE
 6737
 6738  032674  012765  100000  000000  2$:     MOV     #CCLR,RKCS1(R5)
 6739  032702  012737  177777  001366          MOV     #-1,COUNT
 6740  032710  004737  047114                  JSR     PC,CLKON
 6741  032714  013704  001222                  MOV     $UNIT,R4
 6742  032720  063704  005464                  ADD     UNITB,R4
 6743  032724  004737  044214                  JSR     PC,FATT3
 6744  032730  104110                          ERROR   110             ;NO ATTN ON PORT A TO ALLOW SEIZE
 6745
 6746  032732  004737  047162                  JSR     PC,CLKOF        ;TURN CLOCK OFF
 6747  032736  005137  001366                  COM     COUNT           ;GET ACTUAL COUNT OF TIMEOUT
 6748  032742  023727  001366  000043          CMP     COUNT,#35.      ;COMPARE AGAINST APPROX 1 SEC
 6749  032750  002001                          BGE     TST31           ;;GO TO NEXT TST
 6750  032752  104153                          ERROR   153             ;TIMEOUT DID NOT RE-TRIGGER
 6751
 6752                                  ;;**********************************************************************
 6753                                  ;*TEST 31        TEST PORT 'B' TIMER INHIBIT
 6754                                  ;*
 6755                                  ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
 6756                                  ;*
 6757                                  ;;**********************************************************************
 6758  032754  000004                  TST31:  SCOPE
 6759  032756  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 6760  032764  012706  001100                  MOV     #STACK,SP
 6761  032770  004737  045534                  JSR     PC,SUBCLR
 6762  032774  104024                          ERROR   24              ;CERR AFTER SCLR
 6763  032776  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
 6764  033004  012737  000000  005464          MOV     #0,UNITB
 6765  033012  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6766  033020  112737  000101  056576          MOVB    #'A,MSG19A
 6767  033026  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
 6768  033034  012737  000001  005314          MOV     #SELDRV,HCS1
 6769  033042  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 6770  033046  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
 6771
 6772  033050  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
 6773  033056  012737  000001  005464          MOV     #1,UNITB
```

```
 6774  033064  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6775  033072  112737  000102  056576          MOVB    #'B,MSG19A
 6776  033100  012737  000001  005314          MOV     #SELDRV,HCS1
 6777  033106  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 6778  033112  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
 6779
 6780  033114  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
 6781  033122  001001                          BNE     64$             ;BR IF YES
 6782  033124  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
 6783  033126                          64$:
 6784  033126  004737  045534                  JSR     PC,SUBCLR
 6785  033132  104024                          ERROR   24              ;CERR AFTER SCLR
 6786
 6787  033134  012737  000000  005464          MOV     #0,UNITB        ;SETUP FOR PORT A
 6788  033142  112737  000101  056576          MOVB    #'A,MSG19A
 6789  033150  004737  044172                  JSR     PC,DRAV         ;PORT A TRIES TO SEIZE THE DRIVE
 6790  033154  000401                          BR      1$              ;BR IF NOT AVAIL
 6791  033156  104103                          ERROR   103             ;PORT A AVAIL BEFORE TMO OR RELEASE
 6792
 6793  033160  012765  100000  000000  1$:     MOV     #CCLR,RKCS1(R5)
 6794  033166  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
 6795  033174  012737  000001  005464          MOV     #1,UNITB
 6796  033202  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6797  033210  112737  000102  056576          MOVB    #'B,MSG19A
 6798  033216  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
 6799  033224  012737  000001  005314          MOV     #SELDRV,HCS1
 6800  033232  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 6801  033236  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
 6802
 6803  033240  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
 6804  033246  012737  000000  005464          MOV     #0,UNITB
 6805  033254  063765  005464  000010          ADD     UNITB,RKCS2(R5)
 6806  033262  112737  000101  056576          MOVB    #'A,MSG19A
 6807  033270  012737  000001  005314          MOV     #SELDRV,HCS1
 6808  033276  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 6809  033302  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
 6810
 6811  033304  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
 6812  033312  001001                          BNE     65$             ;BR IF YES
 6813  033314  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
 6814  033316                          65$:
 6815  033316  012737  000001  005464          MOV     #1,UNITB        ;SETUP FOR PORT B
 6816  033324  112737  000102  056576          MOVB    #'B,MSG19A
 6817  033332  004737  044172                  JSR     PC,DRAV
 6818  033336  000401                          BR      2$
 6819  033340  104103                          ERROR   103             ;PORT B AVAIL BEFORE TMO OR RELEASE
 6820
 6821  033342  012765  100000  000000  2$:     MOV     #CCLR,RKCS1(R5)
 6822  033350  012737  177777  001366          MOV     #-1,COUNT
 6823  033356  004737  047114                  JSR     PC,CLKON
 6824  033362  013704  001222                  MOV     $UNIT,R4
 6825  033366  063704  005464                  ADD     UNITB,R4
 6826  033372  004737  044214                  JSR     PC,FATT3
 6827  033376  104110                          ERROR   110             ;NO ATTN ON PORT B TO ALLOW SEIZE
 6828
 6829  033400  004737  047162                  JSR     PC,CLKOF        ;TURN CLOCK OFF
```

```
 6830  033404  005137  001366                  COM     COUNT           ;GET ACTUAL COUNT OF TIMEOUT
 6831  033410  023727  001366  000043          CMP     COUNT,#35.      ;COMPARE AGAINST APPROX 1 SEC
 6832  033416  002001                          BGE     TST32           ;;GO TO NEXT TST
 6833  033420  104153                          ERROR   153             ;TIMEOUT DID NOT RE-TRIGGER
 6834
```

```
                                        ;;**********************************************************************
                                        ;*TEST 32       TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'A'
                                        ;*
                                        ;*      VERIFY THAT THE UNLOAD COMMAND THRU A PORT, SEIZES THAT
                                        ;*      PORT FOR AS LONG HAS HEADS ARE UNLOADED & RELEASE IS NOT
                                        ;*      ISSUED.
                                        ;*
                                        ;*      A.  ISSUE AN UNLOAD COMMAND THRU PORT 'A'.
                                        ;*          VERIFY DRIVE UNLOADS & ATTENTION IS SET.
                                        ;*
                                        ;*      B.  DELAY FOR 5 SECONDS & VERIFY DRIVE NOT AVAILABLE
                                        ;*          TO PORT 'B' TO INSURE TIMERS INHIBITED
                                        ;*
                                        ;*      C.  ISSUE A RELEASE FROM PORT 'A'.  VERIFY DRIVE BECOMES
                                        ;*          AVAILABLE TO PORT 'B'
                                        ;*
                                        ;*      D.  LOAD HEADS FROM PORT 'B' & VERIFY 'ATTN-B' AT COMPLETION
                                        ;*
                                        ;;**********************************************************************
033422  000004                  TST32:  SCOPE
033424  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
033432  004737  045534                  JSR     PC,SUBCLR       ;SUB SYSTEM CLEAR 29-SEP-77
033436  104024                          ERROR   24
033440  012706  001100                  MOV     #STACK,SP
033444  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
033452  012737  000001  005464          MOV     #1,UNITB
033460  063765  005464  000010          ADD     UNITB,RKCS2(R5)
033466  112737  000102  056576          MOVB    #'B,MSG19A
033474  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
033502  012737  000001  005314          MOV     #SELDRV,HCS1
033510  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
033514  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
                                        
033516  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
033524  012737  000000  005464          MOV     #0,UNITB
033532  063765  005464  000010          ADD     UNITB,RKCS2(R5)
033540  112737  000101  056576          MOVB    #'A,MSG19A
033546  012737  000001  005314          MOV     #SELDRV,HCS1
033554  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
033560  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
                                        
033562  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
033570  001001                          BNE     64$             ;BR IF YES
033572  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
033574                          64$:
033574  004737  045534                  JSR     PC,SUBCLR
033600  104024                          ERROR   24              ;CERR AFTER SCLR
                                        
033602  005237  005276                  INC     UNLD            ;USED FOR VALID HALT
033606  012737  034046  001176          MOV     #2$,$ESCAPE
033614  012737  000007  005314          MOV     #UNLOAD,HCS1    ;UNLOAD CMD
033622  053737  001170  005314          BIS     $TMP4,HCS1
033630  013765  005314  000000          MOV     HCS1,RKCS1(R5)
033636  013737  001400  005352          MOV     T10,TEMP1       ;SETUP TIMEOUT
```

```
6891  033644  004737  044106          JSR     PC,FATT2        ;FIND ATTN
6892  033650  104073                  ERROR   73              ;NO ATTN AFTER UNLD CMD
6893
6894  033652  012737  000454  001366  MOV     #300.,COUNT
6895  033660  004737  047204          JSR     PC,TMO          ;DO 5 SEC DELAY
6896  033664  012737  000001  005464  MOV     #1,UNITB        ;SETUP FOR PORT B
6897  033672  112737  000102  056576  MOVB    #'B,MSG19A
6898  033700  004737  044172          JSR     PC,DRAV
6899  033704  000401                  BR      1$
6900  033706  104155                  ERROR   155             ;PORT B AVAIL BEFORE RLS WHEN UNLOADED
6901                                                          ;UNLOAD DID NOT INHIBIT TIMERS
6902
6903  033710  012765  100000  000000  1$:     MOV     #CCLR,RKCS1(R5)
6904  033716  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6905  033724  012737  000000  005464          MOV     #0,UNITB
6906  033732  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6907  033740  112737  000101  056576          MOVB    #'A,MSG19A
6908  033746  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6909  033754  012737  000001  005314          MOV     #SELDRV,HCS1
6910  033762  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6911  033766  104117                  ERROR   117             ;NO RDY AFTER SEL DRV CMD
6912
6913  033770  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6914  033776  012737  000001  005464          MOV     #1.UNITB
6915  034004  063765  005464  000010          ADD     UNITB,RKCS2(R5)
6916  034012  112737  000102  056576          MOVB    #'B,MSG19A
6917  034020  012737  000001  005314          MOV     #SELDRV,HCS1
6918  034026  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6919  034032  104117                  ERROR   117             ;NO RKY AFTER SEL DRV CMD
6920
6921  034034  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6922  034042  001001                          BNE     65$             ;BR IF YES
6923  034044  104071                  ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6924  034046                          65$:
6925  034046  005037  001176          2$:     CLR     $ESCAPE
6926
6927  034052  004737  045534                  JSR     PC,SUBCLR
6928  034056  104024                  ERROR   24              ;CERR AFTER SCLR
6929
6930  034060  012737  000011  005314          MOV     #SRTSPL,HCS1
6931  034066  004737  043372                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
6932  034072  104121                  ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD
6933
6934  034074  013737  001406  005354          MOV     T100,TEMP2      ;SETUP TIMEOUT
6935  034102  004737  044006                  JSR     PC,FATT1        ;FIND ATTN
6936  034106  104074                  ERROR   74              ;NO ATTN AFTER ST SPIN CMD
6937
6938  034110  005037  005276                  CLR     UNLD
6939
6940  034114  005037  005276                  CLR     UNLD
6941
6942                                  ;*********************************************************************
6943                                  ;*TEST 33       TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'B'
6944                                  ;*
6945                                  ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'
6946                                  ;*
```

```
6947                                    ;;**********************************************************************
6948  034120  000004                    TST33:  SCOPE
6949  034122  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
6950  034130  004737  045534                    JSR     PC,SUBCLR       ;SUB SYSTEM CLEAR 29-SEP-77
6951  034134  104024                            ERROR   24
6952  034136  012706  001100                    MOV     #STACK,SP
6953  034142  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6954  034150  012737  000000  005464            MOV     #0,UNITB
6955  034156  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6956  034164  112737  000101  056576            MOVB    #'A,MSG19A
6957  034172  062765  000010  000010            ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
6958  034200  012737  000001  005314            MOV     #SELDRV,HCS1
6959  034206  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6960  034212  104117                            ERROR   117             ;NO RDY AFTER SEL DRV CMD
6961
6962  034214  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6963  034222  012737  000001  005464            MOV     #1,UNITB
6964  034230  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6965  034236  112737  000102  056576            MOVB    #'B,MSG19A
6966  034244  012737  000001  005314            MOV     #SELDRV,HCS1
6967  034252  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6968  034256  104117                            ERROR   117             ;NO RKY AFTER SEL DRV CMD
6969
6970  034260  032737  000040  005342            BIT     #D DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6971  034266  001001                            BNE     64$             ;BR IF YES
6972  034270  104071                            ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
6973  034272                          64$:
6974  034272  004737  045534                    JSR     PC,SUBCLR
6975  034276  104024                            ERROR   24              ;CERR AFTER SCLR
6976
6977  034300  005237  005276                    INC     UNLD            ;USED FOR VALID HALT
6978  034304  012737  034544  001176            MOV     #2$,$ESCAPE
6979  034312  012737  000007  005314            MOV     #UNLOAD,HCS1    ;UNLOAD CMD
6980  034320  053737  001170  005314            BIS     $TMP4,HCS1
6981  034326  013765  005314  000000            MOV     HCS1,RKCS1(R5)
6982  034334  013737  001400  005352            MOV     T10,TEMP1       ;SETUP TIMEOUT
6983  034342  004737  044106                    JSR     PC,FATT2        ;FIND ATTN
6984  034346  104073                            ERROR   73              ;NO ATTN AFTER UNLD CMD
6985
6986  034350  012737  000454  001366            MOV     #300.,COUNT
6987  034356  004737  047204                    JSR     PC,TMO          ;DO 5 SEC DELAY
6988  034362  012737  000000  005464            MOV     #0,UNITB        ;SETUP FOR PORT A
6989  034370  112737  000101  056576            MOVB    #'A,MSG19A
6990  034376  004737  044172                    JSR     PC,DRAV
6991  034402  000401                            BR      1$
6992  034404  104155                            ERROR   155             ;PORT A AVAIL BEFORE RLS WHEN UNLOADED
6993                                            ;UNLOAD DID NOT INHIBIT TIMERS
6994
6995  034406  012765  100000  000000  1$:       MOV     #CCLR,RKCS1(R5)
6996  034414  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6997  034422  012737  000001  005464            MOV     #1,UNITB
6998  034430  063765  005464  000010            ADD     UNITB,RKCS2(R5)
6999  034436  112737  000102  056576            MOVB    #'B,MSG19A
7000  034444  062765  000010  000010            ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
7001  034452  012737  000001  005314            MOV     #SELDRV,HCS1
7002  034460  004737  043372                    JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
```

```
7003  034464  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
7004
7005  034466  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7006  034474  012737  000000  005464          MOV     #0,UNITB
7007  034502  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7008  034510  112737  000101  056576          MOVB    #'A,MSG19A
7009  034516  012737  000001  005314          MOV     #SELDRV,HCS1
7010  034524  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7011  034530  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
7012
7013  034532  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
7014  034540  001001                          BNE     65$             ;BR IF YES
7015  034542  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
7016  034544                          65$:
7017  034544  005037  001176          2$:     CLR     $ESCAPE
7018
7019  034550  004737  045534                  JSR     PC,SUBCLR
7020  034554  104024                          ERROR   24              ;CERR AFTER SCLR
7021
7022  034556  012737  000011  005314          MOV     #SRTSPL,HCS1
7023  034564  004737  043372                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
7024  034570  104121                          ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.
7025
7026  034572  013737  001406  005354          MOV     T100,TEMP2      ;SETUP TIMEOUT
7027  034600  004737  044006                  JSR     PC,FATT1        ;FIND ATTN
7028  034604  104074                          ERROR   74              ;NO ATTN AFTER ST SPIN CMD.
7029
7030  034606  005037  005276                  CLR     UNLD
7031
7032  034612  005037  005276                  CLR     UNLD
7033
7034
7035                                  ;;**********************************************************************
7036                                  ;*TEST 34        TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'A'
7037                                  ;*
7038                                  ;*      VERIFY THAT THE RECAL COMMAND THRU A PORT SEIZES THAT PORT
7039                                  ;*      FOR AS LONG AS THE RECAL IS IN PROGRESS & RELEASE IS NOT ISSUED.
7040                                  ;*
7041                                  ;*      A.      ISSUE A RECAL COMMAND FROM CYL 410 THRU PORT 'A'
7042                                  ;*
7043                                  ;*      B.      VERIFY PORT 'B' CANNOT SEIZE THE DRIVE UNTIL PORT 'A'
7044                                  ;*              RECEIVES ATTN AND TIMES OUT. THIS INSURES THAT
7045                                  ;*              THE TIMERS ARE INHIBITED.
7046                                  :*
7047                                  ;;**********************************************************************
7048  034616  000004                  TST34:  SCOPE
7049  034620  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
7050  034626  004737  045534                  JSR     PC,SUBCLR       ;SUB SYSTEM CLEAR 29-SEP-77
7051  034632  104024                          ERROR   24              ;29-SEP-77
7052  034634  012706  001100                  MOV     #STACK,SP
7053  034640  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7054  034646  012737  000001  005464          MOV     #1,UNITB
7055  034654  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7056  034662  112737  000102  056576          MOVB    #'B,MSG19A
7057  034670  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
7058  034676  012737  000001  005314          MOV     #SELDRV,HCS1
```

```
7059  034704  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7060  034710  104117                  ERROR   117             ;NO RDY AFTER SEL DRV CMD
7061
7062  034712  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7063  034720  012737  000000  005464  MOV     #0,UNITB
7064  034726  063765  005464  000010  ADD     UNITB,RKCS2(R5)
7065  034734  112737  000101  056576  MOVB    #'A,MSG19A
7066  034742  012737  000001  005314  MOV     #SELDRV,HCS1
7067  034750  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7068  034754  104117                  ERROR   117             ;NO RKY AFTER SEL DRV CMD
7069
7070  034756  032737  000040  005342  BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
7071  034764  001001                  BNE     64$             ;BR IF YES
7072  034766  104071                  ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
7073  034770                  64$:
7074  034770  004737  045534          JSR     PC,SUBCLR
7075  034774  104024                  ERROR   24              ;CERR AFTER SCLR
7076
7077  034776  012737  035464  001176  MOV     #5$,$ESCAPE
7078  035004  013765  012770  000020  MOV     LC,RKDC(R5)     ;SEEK TO LAST CYL
7079
7080  035012  012737  000017  005314  MOV     #SEEK,HCS1
7081  035020  004737  043372          JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
7082  035024  104131                  ERROR   131             ;NO RDY AFTER SEEK CMD
7083
7084  035026  013737  001412  005352  MOV     T50000,TEMP1    ;SETUP TIMEOUT
7085  035034  004737  044106          JSR     PC,FATT2        ;FIND ATTN
7086  035040  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
7087
7088  035042  032737  100000  005314  BIT     #CERR,HCS1
7089  035050  001401                  BEQ     65$
7090  035052  104210                  ERROR   210             ;CERR AFTER SEEK CMD
7091
7092  035054                  65$:
7093
7094  035054  012737  050340  005404  MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7095  035062  005037  005406          CLR     E.B0            ;EXPECTED MSG B0
7096  035066  012737  001720  005410  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7097  035074  012737  000001  005412  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7098  035102  005037  005414          CLR     E.A2            ;EXPECTED MSG A2
7099  035106  012737  000002  005416  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7100  035114  012737  000003  005422  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7101
7102  035122  004737  044274          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7103  035126  000000                   WORD    0!0!0           ;& MSGS SPECIFIED HERE
7104  035130  104161                  ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
7105  035132  104162                  ERROR   162             ;MSH B0 ERROR
7106  035134  104163                  ERROR   163             ;MSG A1 ERROR
7107  035136  104164                  ERROR   164             ;MSG B1 ERROR
7108
7109
7110  035140  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
7111  035146  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
7112  035154  063765  005464  000010  ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
7113  035162  012737  000005  005314  MOV     #CLEAR,HCS1
7114  035170  004737  043372          JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
```

```
 7115  035174  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 7116  035176  004737  043750                  JSR     PC,TSTATN       ;TEST FOR ATTN
 7117  035202  000401                          BR      66$
 7118  035204  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 7119  035206                          66$:
 7120
 7121  035206  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 7122  035214  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
 7123  035220  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 7124  035226  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 7125  035234  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
 7126  035240  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 7127  035246  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 7128
 7129  035254  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 7130  035260  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 7131  035262  104033                          ERROR   33              ;MSG A0 ERROR AFTER DRV CLEAR CMD
 7132  035264  104034                          ERROR   34              ;MSH B0 ERROR
 7133  035266  104035                          ERROR   35              ;MSG A1 ERROR
 7134  035270  104036                          ERROR   36              ;MSG B1 ERROR
 7135
 7136  035272  012737  000013  005314          MOV     #RECAL,HCS1     ;RECAL COMMAND
 7137  035300  053737  001170  005314          BIS     $TMP4,HCS1
 7138  035306  013765  005314  000000          MOV     HCS1,RKCS1(R5)
 7139  035314  013704  001222                  MOV     $UNIT,R4
 7140  035320  063704  005464                  ADD     UNITB,R4
 7141  035324  136465  005304  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5)   ;SEE IF ATTN SET
 7142  035332  001027                          BNE     3$              ;BR IF YES
 7143
 7144  035334  012737  000001  005464          MOV     #1,UNITB        ;SETUP FOR PORT B
 7145  035342  112737  000102  056576          MOVB    #'B,MSG19A
 7146  035350  004737  044172                  JSR     PC,DRAV         ;SEE IF DRV AVAIL
 7147  035354  000401                          BR      2$              ;RETURN HERE IF NO
 7148  035356  104177                          ERROR   177             ;PORT B AVAIL
 7149                                                                  ;RECAL DID NOT INHIBIT TIMERS
 7150  035360  012765  100000  000000  2$:     MOV     #CCLR,RKCS1(R5)
 7151  035366  012737  000000  005464          MOV     #0,UNITB
 7152  035374  112737  000101  056576          MOVB    #'A,MSG19A      ;SETUP PORT A
 7153  035402  004737  044172                  JSR     PC,DRAV         ;SEE IF DRV AVAIL
 7154  035406  104203                          ERROR   203             ;PORT A NOT REMAIN AVAIL DURING RECAL
 7155  035410  000745                          BR      1$
 7156
 7157  035412  004737  045146          3$:     JSR     PC,GSTAT
 7158  035416  032737  020000  005342          BIT     #D.PIP,HMR2     ;SEE IF ANY MOTION
 7159  035424  001401                          BEQ     4$              ;BR IF NO
 7160  035426  104072                          ERROR   72              ;PIP SET AFTER ATTN RECB FROM RECAL
 7161  035430  012737  000170  001366  4$:     MOV     #120.,COUNT
 7162  035436  004737  047204                  JSR     PC,TMO          ;DO 2 SEC DLY
 7163  035442  012737  000001  005464          MOV     #1,UNITB        ;SETUP FOR PORT B
 7164  035450  112737  000102  056576          MOVB    #'B,MSG19A
 7165  035456  004737  044172                  JSR     PC,DRAV         ;SEE IF DRV NOW AVAIL
 7166  035462  104045                          ERROR   45              ;PORT B NOT AVAIL AFTER TMO
 7167
 7168  035464  005037  001176          5$:     CLR     $ESCAPE
 7169
 7170                          ;;*****************************************************************
```

```
                                        ;*TEST 35       TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'B'
                                        ;*
                                        ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
                                        ;*
                                        ;;***********************************************************
035470  000004                          TST35:  SCOPE
035472  012737  000001  001174                  MOV     #1,$TIMES               ;;DO 1 ITERATION
035500  004737  045534                          JSR     PC,SUBCLR               ;SUBSYSTEM CLEAR 29-SEP-77
035504  104024                                  ERROR   24                      ;29-SEP-77
035506  012706  001100                          MOV     #STACK,SP
035512  013765  001222  000010                  MOV     $UNIT,RKCS2(R5)         ;SETUP FOR PORT A
035520  012737  000000  005464                  MOV     #0,UNITB
035526  063765  005464  000010                  ADD     UNITB,RKCS2(R5)
035534  112737  000101  056576                  MOVB    #'A,MSG19A
035542  062765  000010  000010                  ADD     #RLS,RKCS2(R5)          ;RELEASE PORT A
035550  012737  000001  005314                  MOV     #SELDRV,HCS1
035556  004737  043372                          JSR     PC,DOCMD                ;DO SELDRV (STATUS) CMD & GET CONTR RDY
035562  104117                                  ERROR   117                     ;NO RDY AFTER SEL DRV CMD

035564  013765  001222  000010                  MOV     $UNIT,RKCS2(R5)         ;SETUP FOR PORT B
035572  012737  000001  005464                  MOV     #1,UNITB
035600  063765  005464  000010                  ADD     UNITB,RKCS2(R5)
035606  112737  000102  056576                  MOVB    #'B,MSG19A
035614  012737  000001  005314                  MOV     #SELDRV,HCS1
035622  004737  043372                          JSR     PC,DOCMD                ;DO SELDRV (STATUS) CMD & GET CONTR RDY
035626  104117                                  ERROR   117                     ;NO RKY AFTER SEL DRV CMD

035630  032737  000040  005342                  BIT     #D.DRA,HMR2             ;SEE IF DRIVE AVAIL ON PORT B
035636  001001                                  BNE     64$                     ;BR IF YES
035640  104071                                  ERROR   71                      ;PORT B NOT AVAIL AFTER PORT A RLS
035642                                  64$:
035642  004737  045534                          JSR     PC,SUBCLR
035646  104024                                  ERROR   24                      ;CERR AFTER SCLR

035650  012737  036336  001176                  MOV     #5$,$ESCAPE
035656  013765  012770  000020                  MOV     LC,RKDC(R5)             ;SEEK TO LAST CYL

035664  012737  000017  005314                  MOV     #SEEK,HCS1
035672  004737  043372                          JSR     PC,DOCMD                ;DO SEEK CMD & GET CONTR READY
035676  104131                                  ERROR   131                     ;NO RDY AFTER SEEK CMD

035700  013737  001412  005352                  MOV     T50000,TEMP1            ;SETUP TIMEOUT
035706  004737  044106                          JSR     PC,FATT2                ;FIND ATTN
035712  104132                                  ERROR   132                     ;NO ATTN AFTER SEEK CMD

035714  032737  100000  005314                  BIT     #CERR,HCS1
035722  001401                                  BEQ     65$
035724  104210                                  ERROR   210                     ;CERR AFTER SEEK CMD

035726                                  65$:

035726  012737  050340  005404                  MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
035734  005037  005406                          CLR     E.B0                    ;EXPECTED MSG B0
035740  012737  001720  005410                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
035746  012737  000001  005412                  MOV     #1,E.B1                 ;MSG ID FOR EXPECTED MSG B1
035754  005037  005414                          CLR     E.A2                    ;EXPECTED MSG A2
```

```
7227  035760  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
7228  035766  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
7229
7230  035774  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
7231  036000  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
7232  036002  104161                          ERROR   161              ;MSG A0 ERROR AFTER SEEK CMD
7233  036004  104162                          ERROR   162              ;MSH B0 ERROR
7234  036006  104163                          ERROR   163              ;MSG A1 ERROR
7235  036010  104164                          ERROR   164              ;MSG B1 ERROR
7236
7237
7238  036012  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7239  036020  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7240  036026  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
7241  036034  012737  000005  005314          MOV     #CLEAR,HCS1
7242  036042  004737  043372                  JSR     PC,DOCMD         ;DO DRIVE CLEAR CMD & GET CONTR RDY
7243  036046  104151                          ERROR   151              ;NO RDY AFTER DRIVE CLEAR CMD
7244  036050  004737  043750                  JSR     PC,TSTATN        ;TEST FOR ATTN
7245  036054  000401                          BR      66$
7246  036056  104154                          ERROR   154              ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7247  036060                          66$:
7248
7249  036060  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7250  036066  005037  005406                  CLR     E.B0             ;EXPECTED MSG B0
7251  036072  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7252  036100  012737  000001  005412          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
7253  036106  005037  005414                  CLR     E.A2             ;EXPECTED MSG A2
7254  036112  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
7255  036120  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
7256
7257  036126  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
7258  036132  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
7259  036134  104033                          ERROR   33               ;MSG A0 ERROR AFTER DRV CLEAR CMD
7260  036136  104034                          ERROR   34               ;MSH B0 ERROR
7261  036140  104035                          ERROR   35               ;MSG A1 ERROR
7262  036142  104036                          ERROR   36               ;MSG B1 ERROR
7263
7264  036144  012737  000013  005314          MOV     #RECAL,HCS1      ;RECAL COMMAND
7265  036152  053737  001170  005314          BIS     $TMP4,HCS1
7266  036160  013765  005314  000000          MOV     HCS1,RKCS1(R5)
7267  036166  013704  001222                  MOV     $UNIT,R4
7268  036172  063704  005464                  ADD     UNITB,R4
7269  036176  136465  005304  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5)   ;SEE IF ATTN SET
7270  036204  001027                          BNE     3$               ;BR IF YES
7271
7272  036206  012737  000000  005464          MOV     #0,UNITB         ;SETUP FOR PORT A
7273  036214  112737  000101  056576          MOVB    #'A,MSG19A
7274  036222  004737  044172                  JSR     PC,DRAV          ;SEE IF DRV AVAIL
7275  036226  000401                          BR      2$               ;RETURN HERE IF NO
7276  036230  104177                          ERROR   177              ;PORT A AVAIL
7277                                                                   ;RECAL DID NOT INHIBIT TIMERS
7278  036232  012765  100000  000000  2$:     MOV     #CCLR,RKCS1(R5)
7279  036240  012737  000001  005464          MOV     #1,UNITB
7280  036246  112737  000102  056576          MOVB    #'B,MSG19A       ;SETUP PORT B
7281  036254  004737  044172                  JSR     PC,DRAV          ;SEE IF DRV AVAIL
7282  036260  104203                          ERROR   203              ;PORT B NOT REMAIN AVAIL DURING RECAL
```

```
7283  036262  000745                          BR      1$
7284
7285  036264  004737  045146          3$:     JSR     PC,GSTAT
7286  036270  032737  020000  005342          BIT     #D.PIP,HMR2     ;SEE IF ANY MOTION
7287  036276  001401                          BEQ     4$              ;BR IF NO
7288  036300  104072                          ERROR   72              ;PIP SET AFTER ATTN RECA FROM RECAL
7289  036302  012737  000170  001366  4$:     MOV     #120.,COUNT
7290  036310  004737  047204                  JSR     PC,TMO          ;DO 2 SEC DLY
7291  036314  012737  000000  005464          MOV     #0,UNITB        ;SETUP FOR PORT A
7292  036322  112737  000101  056576          MOVB    #'A,MSG19A
7293  036330  004737  044172                  JSR     PC,DRAV         ;SEE IF DRV NOW AVAIL
7294  036334  104045                          ERROR   45              ;PORT A NOT AVAIL AFTER TMO
7295
7296  036336  005037  001176          5$:     CLR     $ESCAPE
7297
7298
7299                                  ;;**********************************************************************
7300                                  ;*TEST 36       READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
7301                                  ;*
7302                                  ;*      THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
7303                                  ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
7304                                  ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
7305                                  ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
7306                                  ;*
7307                                  ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
7308                                  ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
7309                                  ;*
7310                                  ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
7311                                  ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
7312                                  ;*      A MESSAGE WILL BE TYPED INDICATING THAT ALL
7313                                  ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
7314                                  ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
7315                                  ;*
7316                                  ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
7317                                  ;*
7318                                  ;;**********************************************************************
7319  036342  000004                  TST36:  SCOPE
7320  036344  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
7321  036352  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7322
7323  036356  004737  045534                  JSR     PC,SUBCLR
7324  036362  104024                          ERROR   24              ;CERR AFTER SCLR
7325
7326  036364  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7327  036372  012737  000001  005464          MOV     #1,UNITB
7328  036400  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7329  036406  112737  000102  056576          MOVB    #'B,MSG19A
7330  036414  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
7331  036422  012737  000001  005314          MOV     #SELDRV,HCS1
7332  036430  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7333  036434  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
7334
7335  036436  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7336  036444  012737  000000  005464          MOV     #0,UNITB
7337  036452  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7338  036460  112737  000101  056576          MOVB    #'A,MSG19A
```

```
7339  036466  012737  000001  005314          MOV     #SELDRV,HCS1
7340  036474  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7341  036500  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
7342
7343  036502  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
7344  036510  001001                          BNE     64$             ;BR IF YES
7345  036512  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
7346  036514                          64$:
7347
7348  036514  004737  045534                  JSR     PC,SUBCLR
7349  036520  104024                          ERROR   24              ;CERR AFTER SCLR
7350
7351  036522  005037  005354                  CLR     TEMP2           ;SECTOR CTR
7352  036526  005037  005356                  CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
7353                                                                  ;1=22 SECTOR SOFTWARE DETECTED TABLE
7354                                                                  ;2=DONE
7355  036532  012737  002276  005360          MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE ADDR.
7356  036540  013765  005360  000004          MOV     TEMP4,RKBA(R5)
7357  036546  012737  001000  005362          MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
7358  036554  013765  005362  000006          MOV     TEMP5,RKDA(R5)
7359
7360  036562  013765  012770  000020  1$:     MOV     LC,RKDC(R5)     ;LAST CYL
7361  036570  012765  177400  000002          MOV     #-256.,RKWC(R5) ;LOAD WORD CT
7362  036576  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
7363  036604  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7364  036612  012737  000021  005314          MOV     #RDDATA,HCS1
7365  036620  004737  043430                  JSR     PC,DATCMD       ;DO READ DATA CMD & GET CONTR READY
7366  036624  104226                          ERROR   226             ;NO RDY AFTER READ DATA CMD
7367  036626  004737  045146                  JSR     PC,GSTAT        ;GET FRESH DATA
7368  036632  032737  100000  005314          BIT     #CERR,HCS1
7369  036640  001470                          BEQ     8$
7370  036642  104227                          ERROR   227             ;CERR AFTER READ DATA CMD
7371
7372  036644  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7373  036652  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
7374  036656  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7375  036664  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7376  036672  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
7377  036676  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7378  036704  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7379
7380  036712  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7381  036716  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
7382  036720  104054                          ERROR   54              ;MSG A0 ERROR AFTER READ DATA CMD
7383  036722  104026                          ERROR   26              ;MSH B0 ERROR
7384  036724  104056                          ERROR   56              ;MSG A1 ERROR
7385  036726  104030                          ERROR   30              ;MSG B1 ERROR
7386
7387  036730  004737  045534                  JSR     PC,SUBCLR
7388  036734  104024                          ERROR   24              ;CERR AFTER SUBCLR
7389
7390  036736  005237  005354                  INC     TEMP2
7391  036742  023727  005354  000005          CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
7392  036750  001007                          BNE     5$
7393  036752  005737  005356                  TST     TEMP3
7394  036756  001002                          BNE     2$
```

```
7395  036760  104233                          ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
7396  036762  000414                          BR      3$
7397  036764  104230                  2$:     ERROR   230             ;CANT READ SECTORS 10,12,14,16,18,20
7398  036766  000412                          BR      3$
7399
7400  036770  013765  005360  000004  5$:     MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
7401  036776  062737  000002  005362          ADD     #2,TEMP5        ;SETUP TO READ 2 SECTORS FROM LAST
7402  037004  013765  005362  000006          MOV     TEMP5,RKDA(R5)
7403  037012  000663                          BR      1$
7404
7405  037014  005237  001452          3$:     INC     BSERR           ;SET BSE FLAG
7406  037020  000454                          BR      TST37           ;;GO TO NEXT TEST
7407  037022  005737  002304          8$:     TST     BSE22H+6        ;TEST CARTRIDGE TYPE
7408  037026  001404                          BEQ     9$              ;BRANCH IF DATA CARTRIDGE
7409  037030  104235                          ERROR   235             ;ALIGNMENT CARTRIDGE USED
7410  037032  005237  001452                  INC     BSERR           ;SET BSE ERROR FLAG
7411  037036  000426                          BR      10$
7412
7413  037040  005237  005356          9$:     INC     TEMP3
7414  037044  023727  005356  000001          CMP     TEMP3,#1
7415  037052  001020                          BNE     10$
7416  037054  005037  005354                  CLR     TEMP2
7417  037060  012737  003276  005360          MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
7418  037066  013765  005360  000004          MOV     TEMP4,RKBA(R5)
7419  037074  012737  001012  005362          MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12
7420  037102  013765  005362  000006          MOV     TEMP5,RKDA(R5)
7421  037110  000137  036562                  JMP     1$              ;REPEAT
7422
7423  037114  005737  001216          10$:    TST     $PASS
7424  037120  001014                          BNE     TST37           ;;GO TO NEXT TST IF NOT 1'ST PASS
7425  037122  104401  056537                  TYPE    ,MSG17          ;CART SERIAL #
7426  037126  012746  002276                  MOV     #BSE22H,-(SP)
7427  037132  004737  053660                  JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
7428  037136  004737  054230                  JSR     PC,$SUPRS       ;TYPE SERIAL #
7429  037142  104401  001205                  TYPE    ,$CRLF
7430  037146  104401  001205                  TYPE    ,$CRLF
7431
7432
7433                                  ;;**********************************************************************
7434                                  ;*TEST 37        DATA TESTS
7435                                  ;*
7436                                  ;*      VERIFY UNIQUE DATA CAN BE WRITTEN THRU EITHER PORT & READ
7437                                  ;*      BACK CORRECTLY THRU BOTH PORTS.
7438                                  ;*
7439                                  ;*      A.  ALL 0'S ARE WRITTEN THRU PORT 'A' ON CYL 0, SECTOR 0,
7440                                  ;*          TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.
7441                                  ;*
7442                                  ;*      B.  ALL 1'S ARE WRITTEN THRU PORT 'B' ON CYL 10, SECTOR 0,
7443                                  ;*          TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.
7444                                  ;*
7445                                  ;*      C.  THE PROGRAM CHECKS THAT CYL 0 WAS NOT OVERWRITTEN
7446                                  ;*          BY READING & VERIFING ALL 0'S THRU PORT 'B'.
7447                                  ;*
7448                                  ;;**********************************************************************
7449  037152  000004                  TST37:  SCOPE
7450  037154  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
7451  037162  012706  001100                  MOV     #STACK,SP
7452
7453  037166  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7454  037174  012737  000001  005464          MOV     #1,UNITB
7455  037202  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7456  037210  112737  000102  056576          MOVB    #'B,MSG19A
7457  037216  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
7458  037224  012737  000001  005314          MOV     #SELDRV,HCS1
7459  037232  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7460  037236  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
7461
7462  037240  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7463  037246  012737  000000  005464          MOV     #0,UNITB
7464  037254  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7465  037262  112737  000101  056576          MOVB    #'A,MSG19A
7466  037270  012737  000001  005314          MOV     #SELDRV,HCS1
7467  037276  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7468  037302  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
7469
7470  037304  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
7471  037312  001001                          BNE     64$             ;BR IF YES
7472  037314  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
7473  037316                          64$:
7474  037316  004737  045534                  JSR     PC,$$BCLR
7475  037322  104024                          ERROR   24              ;CERR AFTER SCLR
7476
7477  037324  005037  001374                  CLR     SECTOR
7478  037330  005037  001360                  CLR     CYLADD
7479  037334  012737  001434  001436          MOV     #DATA0,DATA01           ;HOLD DATA
7480  037342  013765  001374  000006  1$:     MOV     SECTOR,RKDA(R5) ;SETUP TO WRITE
7481  037350  013765  001436  000004          MOV     DATA01,RKBA(R5) ;SETUP DATA
7482  037356  013765  001360  000020          MOV     CYLADD,RKDC(R5) ;SETUP CYLINDER
7483  037364  052765  000020  000010          BIS     #BAI,RKCS2(R5)
7484  037372  012765  177400  000002          MOV     #-256.,RKWC(R5)
7485
7486  037400  012737  000023  005314          MOV     #<WRDATA>,HCS1
7487  037406  004737  043430                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
7488  037412  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
7489  037414  004737  045146                  JSR     PC,GSTAT        ;GET FRESH STATUS
7490  037420  032737  100000  005314          BIT     #CERR,HCS1
7491  037426  001465                          BEQ     68$             ;BR IF NO ERRORS
7492
7493  037430  032737  000200  005330          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
7494  037436  001421                          BEQ     66$             ;BR IF NO
7495  037440  004737  047002                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
7496  037444  000455                          BR      67$             ;RETURN HERE IF NO
7497
7498  037446  005237  001374                  INC     SECTOR          ;RETURN HERE IF YES
7499  037452  023727  001374  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
7500  037460  001003                          BNE     65$             ;BR IF NO
7501  037462  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
7502  037464  000137  040246                  JMP     7$              ;BYPASS TEST
7503  037470  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
7504  037476  000137  037342                  JMP     1$
7505  037502  104012                  66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
7506
```

```
7507  037504  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7508  037512  005037  005406                  CLR     E.B0              ;EXPECTED MSG B0
7509  037516  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7510  037524  012737  000001  005412          MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
7511  037532  005037  005414                  CLR     E.A2              ;EXPECTED MSG A2
7512  037536  012737  000002  005416          MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
7513  037544  012737  000003  005422          MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
7514
7515  037552  004737  044274                  JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
7516  037556  000003                          .WORD   T.A2!T.B2!0               ;& MSGS SPECIFIED HERE
7517  037560  104052                          ERROR   52                ;MSG A0 ERROR AFTER WRITE DATA CMD
7518  037562  104023                          ERROR   23                ;MSH B0 ERROR
7519  037564  104053                          ERROR   53                ;MSG A1 ERROR
7520  037566  104025                          ERROR   25                ;MSG B1 ERROR
7521  037570  104401  056625                  TYPE    ,MSG21            ;ABORTING BALANCE OF TESTS
7522  037574  000137  042644                  JMP     $EOP
7523  037600  104043                  67$:    ERROR   43                ;BAD SECTOR NOT LISTED IN TABLE
7524  037602                          68$:
7525
7526  037602  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7527  037610  005037  005406                  CLR     E.B0              ;EXPECTED MSG B0
7528  037614  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7529  037622  012737  000001  005412          MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
7530  037630  005037  005414                  CLR     E.A2              ;EXPECTED MSG A2
7531  037634  012737  000002  005416          MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
7532  037642  012737  000003  005422          MOV     #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
7533
7534  037650  004737  044274                  JSR     PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
7535  037654  000003                          .WORD   T.A2!T.B2!0               ;& MSGS SPECIFIED HERE
7536  037656  104052                          ERROR   52                ;MSG A0 ERROR AFTER WRITE DATA CMD
7537  037660  104023                          ERROR   23                ;MSH B0 ERROR
7538  037662  104053                          ERROR   53                ;MSG A1 ERROR
7539  037664  104025                          ERROR   25                ;MSG B1 ERROR
7540  037666  005000                          CLR     R0                ;CLEAR TO DO PORT A FIRST
7541
7542  037670  005700                  2$:     TST     R0                ;SEE IF DOING PORT 'A'
7543  037672  001056                          BNE     3$                ;BR IF NO
7544  037674  013765  001222  000010          MOV     $UNIT,RKCS2(R5)   ;SETUP FOR PORT B
7545  037702  012737  000001  005464          MOV     #1,UNITB
7546  037710  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7547  037716  112737  000102  056576          MOVB    #'B,MSG19A
7548  037724  062765  000010  000010          ADD     #RLS,RKCS2(R5)    ;RELEASE PORT B
7549  037732  012737  000001  005314          MOV     #SELDRV,HCS1
7550  037740  004737  043372                  JSR     PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7551  037744  104117                          ERROR   117               ;NO RDY AFTER SEL DRV CMD
7552
7553  037746  013765  001222  000010          MOV     $UNIT,RKCS2(R5)   ;SETUP FOR PORT A
7554  037754  012737  000000  005464          MOV     #0,UNITB
7555  037762  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7556  037770  112737  000101  056576          MOVB    #'A,MSG19A
7557  037776  012737  000001  005314          MOV     #SELDRV,HCS1
7558  040004  004737  043372                  JSR     PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7559  040010  104117                          ERROR   117               ;NO RKY AFTER SEL DRV CMD
7560
7561  040012  032737  000040  005342          BIT     #D.DRA,HMR2       ;SEE IF DRIVE AVAIL ON PORT A
7562  040020  001001                          BNE     69$               ;BR IF YES
```

```
7563  040022  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
7564  040024                          69$:
7565  040024  000137  040246                  JMP     7$              ;GO & WRITE CHECK THRU PORT 'A'
7566
7567  040030  020027  000001          3$:     CMP     R0,#1           ;SEE IF DOING PORT 'B'
7568  040034  001056                          BNE     4$              ;BR IF NO
7569  040036  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7570  040044  012737  000000  005464          MOV     #0,UNITB
7571  040052  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7572  040060  112737  000101  056576          MOVB    #'A,MSG19A
7573  040066  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
7574  040074  012737  000001  005314          MOV     #SELDRV,HCS1
7575  040102  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7576  040106  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
7577
7578  040110  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7579  040116  012737  000001  005464          MOV     #1,UNITB
7580  040124  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7581  040132  112737  000102  056576          MOVB    #'B,MSG19A
7582  040140  012737  000001  005314          MOV     #SELDRV,HCS1
7583  040146  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7584  040152  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
7585
7586  040154  032737  000040  005342          BIT     #D DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
7587  040162  001001                          BNE     70$             ;BR IF YES
7588  040164  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A RLS
7589  040166                          70$:
7590  040166  000137  040246                  JMP     7$              ;GO WRITE CHECK THRU PORT 'B'
7591
7592  040172  020027  000002          4$:     CMP     R0,#2           ;SEE IF DID WRT CHK THRU PORT A & B
7593  040176  001402                          BEQ     5$              ;BR IF YES
7594  040200  000137  040562                  JMP     8$              ;ELSE R0=3, EXIT TEST
7595
7596  040204  023727  001436  001434  5$:     CMP     DATA01,#DATA0   ;SEE IF JUST WROTE 0'S ON CYL 0
7597  040212  001010                          BNE     6$              ;BR IF NO
7598  040214  012737  000012  001360          MOV     #10.,CYLADD     ;ELSE WRITE 1'S ON CYL 10
7599  040222  012737  001440  001436          MOV     #DATA1,DATA01
7600  040230  000137  037342                  JMP     1$              ;GO DO IT
7601
7602  040234  005037  001360          6$:     CLR     CYLADD          ;RECHK CYL 0 FOR 0'S
7603  040240  012737  001434  001436          MOV     #DATA0,DATA01
7604
7605  040246  052765  000020  000010  7$:     BIS     #BAI,RKCS2(R5)  ;SETUP TO WRITE CHECK
7606  040254  013765  001436  000004          MOV     DATA01,RKBA(R5)
7607  040262  013765  001360  000020          MOV     CYLADD,RKDC(R5)
7608  040270  012765  177400  000002          MOV     #-256.,RKWC(R5)
7609  040276  013765  001374  000006          MOV     SECTOR,RKDA(R5)
7610  040304  013737  001374  001370          MOV     SECTOR,SEC      ;COPY FOR FUTURE USE
7611
7612  040312  012737  000031  005314          MOV     #<WRTCHK>,HCS1
7613  040320  004737  043430                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
7614  040324  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
7615  040326  004737  045146                  JSR     PC,GSTAT        ;GET FRESH STATUS
7616  040332  032737  100000  005314          BIT     #CERR,HCS1
7617  040340  001453                          BEQ     72$
7618  040342  032737  040000  005316          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
```

```
7619  040350  001410                          BEQ     71$
7620  040352  016537  000024  001414          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
7621  040360  013737  001434  001416          MOV     DATA0,WD2               ;EXPECTED WORD FOR TYPEOUT
7622  040366  104016                          ERROR   16              ;WCE AFTER WRITE CMD
7623  040370  000437                          BR      72$
7624
7625  040372  104022                  71$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
7626
7627  040374  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7628  040402  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
7629  040406  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7630  040414  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7631  040422  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
7632  040426  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7633  040434  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7634
7635  040442  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7636  040446  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7637  040450  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
7638  040452  104031                          ERROR   31              ;MSH B0 ERROR
7639  040454  104060                          ERROR   60              ;MSG A1 ERROR
7640  040456  104032                          ERROR   32              ;MSG B1 ERROR
7641  040460  104401  056625                  TYPE    ,MSG21          ;ABORTING BALANCE OF TESTS
7642  040464  000137  042644                  JMP     $EOP
7643
7644  040470                          72$:
7645
7646  040470  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7647  040476  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
7648  040502  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7649  040510  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7650  040516  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
7651  040522  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7652  040530  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7653
7654  040536  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7655  040542  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7656  040544  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
7657  040546  104031                          ERROR   31              ;MSH B0 ERROR
7658  040550  104060                          ERROR   60              ;MSG A1 ERROR
7659  040552  104032                          ERROR   32              ;MSG B1 ERROR
7660
7661  040554  005200                          INC     R0
7662  040556  000137  037670                  JMP     2$
7663  040562                          8$:
7664
7665                                  ;;**********************************************************************
7666                                  ;*TEST 40       ALTERNATING SEEK INTERACTION TEST
7667                                  ;*
7668                                  ;*      THIS TEST VERIFIES THAT THERE ARE NO TIMING INTERACTION PROBLEMS
7669                                  ;*      BETWEEN SEEKS FROM BOTH PORTS.
7670                                  ;*
7671                                  ;*      A.  PORT 'A' SEIZES THE DRIVE & SEEKS TO CYLINDER 0 & RELEASES
7672                                  ;*          THE DRIVE AFTER 'ATTN' IS RECEIVED.
7673                                  ;*
7674                                  ;*          THE PROGRAM VERIFIES THAT UNTIL ATTN IS RECEIVED,
```

```
                                        ;*          PORT 'B' SEES CONTROLLER ERROR & DRIVE NOT AVAILABLE.
                                        ;*
                                        ;*      B.  PORT 'B' SEIZES THE DRIVE & SEEKS TO THE LAST CYL
                                        ;*          & RELEASES THE DRIVE AFTER 'ATTN' IS RECEIVED
                                        ;*
                                        ;*          THE PROGRAM VERIFIES THAT UNTIL ATTN IS RECEIVED,
                                        ;*          PORT 'A' SEES CONTROLLER ERROR & DRIVE NOT AVAILABLE.
                                        ;*
                                        ;*      C.  THE ABOVE IS REPEATED FOR A PATTERN OF CONVERGING SEEKS
                                        ;*          TOWARD THE CENTER OF THE CARTRIDGE.
                                        ,*
                                        ;*      D.  THE PROGRAM VERIFIES MULTIPLE ATTENTIONS OR ERRORS
                                        ;*          DO NOT OCCUR AS A RESULT OF TIMING PROBLEMS.
                                        ;*
                                        ;;**********************************************************************
040562  000004                  TST40:  SCOPE
040564  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
040572  012706  001100                  MOV     #STACK,SP
040576  004737  045534                  JSR     PC,SUBCLR
040602  104024                          ERROR   24              ;CERR AFTER SCLR
040604  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
040612  012737  000001  005464          MOV     #1.UNITB
040620  063765  005464  000010          ADD     UNITB,RKCS2(R5)
040626  112737  000102  056576          MOVB    #'B,MSG19A
040634  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
040642  012737  000001  005314          MOV     #SELDRV,HCS1
040650  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
040654  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD

040656  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
040664  012737  000000  005464          MOV     #0,UNITB
040672  063765  005464  000010          ADD     UNITB,RKCS2(R5)
040700  112737  000101  056576          MOVB    #'A,MSG19A
040706  012737  000001  005314          MOV     #SELDRV,HCS1
040714  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
040720  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD

040722  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
040730  001001                          BNE     64$             ;BR IF YES
040732  104071                          ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
040734                          64$:

040734  005037  001414                  CLR     WD1             ;SETUP CONVERGING LIMITS
040740  013737  012770  001416          MOV     LC,WD2

040746  012765  100000  000000  10$:    MOV     #CCLR,RKCS1(R5)
040754  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
040762  063765  005464  000010          ADD     UNITB,RKCS2(R5)
040770  013765  001414  000020          MOV     WD1,RKDC(R5)
040776  012737  000017  005314          MOV     #SEEK,HCS1
041004  004737  043372                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
041010  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD-PORT A

041012  013737  001412  005352          MOV     T50000,TEMP1
```

```
7731  041020  013704  001222          1$:   MOV     $UNIT,R4
7732  041024  136465  005304  000017        BITB    ATTN(R4),RKASOF+1(R5)  ;TEST FOR ATTN ON PORT A
7733  041032  001032                        BNE     3$                     ;BR IF THERE
7734  041034  012737  000001  005464        MOV     #1,UNITB               ;ELSE VERIFY PORT B NOT AVAIL
7735  041042  112737  000102  056576        MOVB    #'B,MSG19A
7736  041050  004737  044172                JSR     PC,DRAV
7737  041054  000403                        BR      2$
7738  041056  104103                        ERROR   103                    ;PORT B AVAIL BEFORE TMO OR RELEASE
7739  041060  000137  042644                JMP     11$
7740
7741  041064  012765  100000  000000  2$:   MOV     #CCLR,RKCS1(R5)
7742  041072  005337  005352                DEC     TEMP1
7743  041076  001350                        BNE     1$                     ;SEE IF PORT A HAS ATTN
7744  041100  005037  005464                CLR     UNITB
7745  041104  112737  000101  056576        MOVB    #'A,MSG19A
7746  041112  104156                        ERROR   156                    ;NO ATTN ON PORT A AFTER SEEK
7747  041114  000137  042644                JMP     11$                    ;EXIT TEST
7748
7749  041120  005037  005464          3$:   CLR     UNITB                  ;SETUP FOR PORT A
7750  041124  112737  000101  056576        MOVB    #'A,MSG19A
7751  041132  004737  044172                JSR     PC,DRAV
7752  041136  104157                        ERROR   157                    ;PORT A NOT AVAIL AFTER SEEK
7753
7754  041140  032737  100000  005314        BIT     #CERR,HCS1
7755  041146  001401                        BEQ     4$
7756  041150  104210                        ERROR   210                    ;CERR AFTER SEEK CMD ON PORT A
7757
7758  041152                          4$:
7759
7760  041152  012737  050340  005404        MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7761  041160  005037  005406                CLR     E.B0                   ;EXPECTED MSG B0
7762  041164  012737  001720  005410        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7763  041172  012737  000001  005412        MOV     #1,E.B1                ;MSG ID FOR EXPECTED MSG B1
7764  041200  005037  005414                CLR     E.A2                   ;EXPECTED MSG A2
7765  041204  012737  000002  005416        MOV     #2,E.B2                ;MSG ID FOR EXPECTED MSG B2
7766  041212  012737  000003  005422        MOV     #3,E.B3                ;MSG ID FOR EXPECTED MSG B3
7767
7768  041220  004737  044274                JSR     PC,CHKMSG              ;CHECK MSGS A0, B0, A1, B1
7769  041224  000000                        .WORD   0!0!0                  ;& MSGS SPECIFIED HERE
7770  041226  104161                        ERROR   161                    ;MSG A0 ERROR AFTER SEEK CMD
7771  041230  104162                        ERROR   162                    ;MSH B0 ERROR
7772  041232  104163                        ERROR   163                    ;MSG A1 ERROR
7773  041234  104164                        ERROR   164                    ;MSG B1 ERROR
7774  041236  013737  001414  001346        MOV     WD1,TOCYL              ;SETUP TO SEE IF ON CORRECT CYL
7775  041244  013765  001414  000020        MOV     WD1,RKDC(R5)
7776  041252  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
7777  041260  063765  005464  000010        ADD     UNITB,RKCS2(R5)
7778
7779
7780  041266  012700  001666                MOV     #RHTAB,R0
7781  041272  012737  000025  005314        MOV     #<RDHEAD>,HCS1
7782  041300  004737  043430                JSR     PC,DATCMD              ;DO DATA X FOR CMD & GET CONTR RDY
7783  041304  104171                        ERROR   171                    ;NO RDY AFTER READ HEADER CMD
7784  041306  032737  100000  005314        BIT     #CERR,HCS1
7785  041314  001401                        BEQ     66$
7786  041316  104174                        ERROR   174                    ;CERR AFTER READ HEADER CMD
```

```
7787
7788  041320  016520  000024          66$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7789  041324  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
7790  041330  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
7791
7792
7793  041334  032765  100000  000010          BIT     #DLT,RKCS2(R5)
7794  041342  001403                          BEQ     67$
7795  041344  004737  045146                  JSR     PC,GSTAT
7796  041350  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
7797  041352                          67$:
7798
7799  041352  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7800  041360  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
7801  041364  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7802  041372  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7803  041400  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
7804  041404  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7805  041412  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7806
7807  041420  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7808  041424  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
7809  041426  104301                          ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
7810  041430  104271                          ERROR   271             ;MSH B0 ERROR
7811  041432  104302                          ERROR   302             ;MSG A1 ERROR
7812  041434  104272                          ERROR   272             ;MSG B1 ERROR
7813
7814  041436  023737  001666  001346          CMP     RHTAB,TOCYL     ;CHECK WORD 0 ONLY, CYL#
7815  041444  001401                          BEQ     65$             ;BR IF SAME
7816  041446  104051                          ERROR   51              ;WRONG CYL# ON HEADER
7817  041450                          65$:
7818
7819  041450  004737  045534                  JSR     PC,SUBCLR
7820  041454  104024                          ERROR   24              ;CERR AFTER SCLR
7821
7822  041456  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7823  041464  012737  000000  005464          MOV     #0,UNITB
7824  041472  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7825  041500  112737  000101  056576          MOVB    #'A,MSG19A
7826  041506  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;RELEASE PORT A
7827  041514  012737  000001  005314          MOV     #SELDRV,HCS1
7828  041522  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7829  041526  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
7830
7831  041530  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7832  041536  012737  000001  005464          MOV     #1,UNITB
7833  041544  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7834  041552  112737  000102  056576          MOVB    #'B,MSG19A
7835  041560  012737  000001  005314          MOV     #SELDRV,HCS1
7836  041566  004737  043372                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7837  041572  104117                          ERROR   117             ;NO RKY AFTER SEL DRV CMD
7838
7839  041574  032737  000040  005342          BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
7840  041602  001001                          BNE     68$             ;BR IF YES
7841  041604  104071                          ERROR   71              ;PORT B NOT AVAIL AFTER PORT A PLS
7842  041606                          68$:
```

```
7843  041606  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7844  041614  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
7845  041622  063765  005464  000010          ADD     UNITB,RKCS2(R5)
7846  041630  013765  001416  000020          MOV     WD2,RKDC(R5)
7847  041636  012737  000017  005314          MOV     #SEEK,HCS1
7848  041644  004737  043372                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
7849  041650  104131                          ERROR   131             ;NO READY AFTER SEEK CMD-PORT B
7850
7851  041652  013737  001412  005352          MOV     T50000,TEMP1
7852  041660  013704  001222          5$:     MOV     $UNIT,R4
7853  041664  005204                          INC     R4
7854  041666  136465  005304  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;TEST FOR ATTN ON PORT B
7855  041674  001032                          BNE     7$              ;BR IF THERE
7856  041676  005037  005464                  CLR     UNITB           ;ELSE VERIFY PORT A NOT AVAIL
7857  041702  112737  000101  056576          MOVB    #'A,MSG19A
7858  041710  004737  044172                  JSR     PC,DRAV
7859  041714  000403                          BR      6$
7860  041716  104103                          ERROR   103             ;PORT A AVAIL BEFORE TMO OR RELEASE
7861  041720  000137  042644                  JMP     11$
7862
7863  041724  012765  100000  000000  6$:     MOV     #CCLR,RKCS1(R5)
7864  041732  005337  005352                  DEC     TEMP1
7865  041736  001350                          BNE     5$              ;SEE IF PORT B HAS ATTN
7866  041740  012737  000001  005464          MOV     #1,UNITB
7867  041746  112737  000102  056576          MOVB    #'B,MSG19A
7868  041754  104156                          ERROR   156             ;NO ATTN ON PORT B AFTER SEEK
7869  041756  000137  042644                  JMP     11$             ;EXIT
7870
7871  041762  012737  000001  005464  7$:     MOV     #1,UNITB        ;SETUP FOR PORT B
7872  041770  112737  000102  056576          MOVB    #'B,MSG19A
7873  041776  004737  044172                  JSR     PC,DRAV
7874  042002  104157                          ERROR   157             ;PORT B NOT AVAIL AFTER SEEK
7875  042004  032737  100000  005314          BIT     #CERR,HCS1
7876  042012  001401                          BEQ     8$
7877  042014  104210                          ERROR   210             ;CERR AFTER SEEK CMD ON PORT B
7878
7879  042016                          8$:
7880
7881  042016  012737  050340  005404          MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7882  042024  005037  005406                  CLR     E.B0            ;EXPECTED MSG B0
7883  042030  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7884  042036  012737  000001  005412          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7885  042044  005037  005414                  CLR     E.A2            ;EXPECTED MSG A2
7886  042050  012737  000002  005416          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7887  042056  012737  000003  005422          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7888
7889  042064  004737  044274                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7890  042070  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
7891  042072  104161                          ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
7892  042074  104162                          ERROR   162             ;MSH B0 ERROR
7893  042076  104163                          ERROR   163             ;MSG A1 ERROR
7894  042100  104164                          ERROR   164             ;MSG B1 ERROR
7895  042102  013737  001416  001346          MOV     WD2,TOCYL       ;SETUP TO SEE IF ON CORRECT CYL
7896  042110  013765  001416  000020          MOV     WD2,RKDC(R5)
7897  042116  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
7898  042124  063765  005464  000010          ADD     UNITB,RKCS2(R5)
```

```


042132  012700  001666                  MOV     #RHTAB,R0
042136  012737  000025  005314          MOV     #<RDHEAD>,HCS1
042144  004737  043430                  JSR     PC,DATCMD           ;DO DATA X FOR CMD & GET CONTR RDY
042150  104171                          ERROR   171                 ;NO RDY AFTER READ HEADER CMD
042152  032737  100000  005314          BIT     #CERR,HCS1
042160  001401                          BEQ     70$
042162  104174                          ERROR   174                 ;CERR AFTER READ HEADER CMD

042164  016520  000024          70$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
042170  016520  000024                  MOV     RKDB(R5),(R0)+      ;2'ND WORD
042174  016520  000024                  MOV     RKDB(R5),(R0)+      ;3'RD WORD


042200  032765  100000  000010          BIT     #DLT,RKCS2(R5)
042206  001403                          BEQ     71$
042210  004737  045146                  JSR     PC,GSTAT
042214  104173                          ERROR   173                 ;DLT AFTER READ HEADER CMD
042216                          71$:

042216  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
042224  005037  005406                  CLR     E.B0                ;EXPECTED MSG B0
042230  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
042236  012737  000001  005412          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
042244  005037  005414                  CLR     E.A2                ;EXPECTED MSG A2
042250  012737  000002  005416          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
042256  012737  000003  005422          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3

042264  004737  044274                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
042270  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
042272  104301                          ERROR   301                 ;MSG A0 ERROR AFTER READ HEADER CMD
042274  104271                          ERROR   271                 ;MSH B0 ERROR
042276  104302                          ERROR   302                 ;MSG A1 ERROR
042300  104272                          ERROR   272                 ;MSG B1 ERROR

042302  023737  001666  001346          CMP     RHTAB,TOCYL         ;CHECK WORD 0 ONLY, CYL#
042310  001401                          BEQ     69$                 ;BR IF SAME
042312  104051                          ERROR   51                  ;WRONG CYL# ON HEADER
042314                          69$:

042314  004737  045534                  JSR     PC,SUBCLR
042320  104024                          ERROR   24                  ;CERR AFTER SCLR

042322  005237  001414                  INC     WD1                 ;CONVERGE THE CYLINDERS
042326  005337  001416                  DEC     WD2
042332  023737  001414  001416          CMP     WD1,WD2             ;SEE IF CONVERGED TOGETHER
042340  001063                          BNE     9$                  ;BR IF NO & REPEAT
042342  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
042350  063765  005464  000010          ADD     UNITB,RKCS2(R5)

042356  012737  000017  005314          MOV     #SEEK,HCS1
042364  004737  043372                  JSR     PC,DOCMD            ;DO SEEK CMD & GET CONTR READY
042370  104131                          ERROR   131                 ;NO RDY AFTER SEEK CMD

042372  013737  001412  005352          MOV     T50000,TEMP1        ;SETUP TIMEOUT
```

```
7955  042400  004737  044106          JSR     PC,FATT2        ;FIND ATTN
7956  042404  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
7957
7958  042406  032737  100000  005314  BIT     #CERR,HCS1
7959  042414  001401                  BEQ     72$
7960  042416  104210                  ERROR   210             ;CERR AFTER SEEK CMD
7961
7962  042420                  72$:
7963
7964  042420  012737  050340  005404  MOV     #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
7965  042426  005037  005406          CLR     E.B0            ;EXPECTED MSG B0
7966  042432  012737  001720  005410  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
7967  042440  012737  000001  005412  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
7968  042446  005037  005414          CLR     E.A2            ;EXPECTED MSG A2
7969  042452  012737  000002  005416  MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
7970  042460  012737  000003  005422  MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
7971
7972  042466  004737  044274          JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
7973  042472  000000                  .WORD   0!0!0           ;& MSGS SPECIFIED HERE
7974  042474  104161                  ERROR   161             ;MSG A0 ERROR AFTER SEEK CMD
7975  042476  104162                  ERROR   162             ;MSH B0 ERROR
7976  042500  104163                  ERROR   163             ;MSG A1 ERROR
7977  042502  104164                  ERROR   164             ;MSG B1 ERROR
7978
7979  042504  000137  042644          JMP     11$             ;EXIT TEST
7980  042510                  9$:
7981  042510  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7982  042516  012737  000001  005464  MOV     #1,UNITB
7983  042524  063765  005464  000010  ADD     UNITB,RKCS2(R5)
7984  042532  112737  000102  056576  MOVB    #'B,MSG19A
7985  042540  062765  000010  000010  ADD     #RLS,RKCS2(R5)  ;RELEASE PORT B
7986  042546  012737  000001  005314  MOV     #SELDRV,HCS1
7987  042554  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7988  042560  104117                  ERROR   117             ;NO RDY AFTER SEL DRV CMD
7989
7990  042562  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7991  042570  012737  000000  005464  MOV     #0,UNITB
7992  042576  063765  005464  000010  ADD     UNITB,RKCS2(R5)
7993  042604  112737  000101  056576  MOVB    #'A,MSG19A
7994  042612  012737  000001  005314  MOV     #SELDRV,HCS1
7995  042620  004737  043372          JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7996  042624  104117                  ERROR   117             ;NO RKY AFTER SEL DRV CMD
7997
7998  042626  032737  000040  005342  BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
7999  042634  001001                  BNE     73$             ;BR IF YES
8000  042636  104071                  ERROR   71              ;PORT A NOT AVAIL AFTER PORT B RLS
8001  042640                  73$:
8002  042640  000137  040746          JMP     10$
8003
8004  042644                  11$:
8005
8006
8007
8008
8009
```

```
                        .SBTTL  END OF PASS ROUTINE

                        ;;**********************************************************************
                        ;*INCREMENT THE PASS NUMBER ($PASS)
                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                        ;*IF THERES A MONITOR GO TO IT
                        ;*IF THERE ISN'T JUMP TO ST5

042644                  $EOP:

042644  000004                 SCOPE
042646  005037  001176         CLR     $ESCAPE
042652  012737  000001  001174 MOV     #1,$TIMES
042660  012706  001100         MOV     #STACK,SP
042664  005237  001220         INC     $DEVCT          ;INCR COUNT FOR # DRIVES CHECKED
042670  023737  005434  001220 CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
042676  001403                 BEQ     $EOP1+2         ;BR IF YES
042700  000137  012472         JMP     NUDRV           ;ELSE TEST NEXT DRIVE PRESENT
042704  000004         $EOP1:  SCOPE
042706  005037  001102         CLR     $TSTNM          ;;ZERO THE TEST NUMBER
042712  005037  001174         CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
042716  005237  001216         INC     $PASS           ;;INCREMENT THE PASS NUMBER
042722  042737  100000  001216 BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
042730  005327                 DEC     (PC)+           ;;LOOP?
042732  000001         $EOPCT: .WORD   1
042734  003022                 BGT     $DOAGN          ;;YES
042736  012737                 MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
042740  000001         $ENDCT: .WORD   1
042742  042732                 $EOPCT
042744  104401  043011         TYPE    ,$ENDMG         ;;TYPE "END PASS #"
042750  013746  001216         MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
042754  104405                 TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
042756  104401  043006         TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
042762  013700  000042 $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
042766  001405                 BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
042770  000005                 RESET                   ;;CLEAR THE WORLD
042772  004710         $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
042774  000240                 NOP                     ;;SAVE ROOM
042776  000240                 NOP                     ;;FOR
043000  000240                 NOP                     ;;ACT11
043002                 $DOAGN:
043002  000137                 JMP     @(PC)+          ;;RETURN
043004  010646         $RTNAD: .WORD   ST5
043006     377     377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
043011     015  042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS #/
043016  050040  051501  020123
043024  000043
```

```
8057                                    .SBTTL  SUBROUTINES
8058
8059                                    ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU UNITB
8060                                    ;
8061
8062  043026  012700  005424           CLRFLG: MOV     #DDUMP,R0
8063  043032  012701  177757                   MOV     #-17.,R1
8064  043036  005020                   1$:     CLR     (R0)+
8065  043040  005201                           INC     R1
8066  043042  001375                           BNE     1$
8067  043044  000207                           RTS     PC
8068
8069                                    ;
8070                                    ;TYPE PROGRAM ID IF FTITLE=0
8071                                    ;
8072
8073  043046  005737  001340           TITLE:  TST     FTITLE
8074  043052  001024                           BNE     1$
8075  043054  005237  001340                   INC     FTITLE
8076  043060  104401  054566                   TYPE    ,MSG1                   ;PROGRAM ID
8077                                    .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
8078  043064  005737  000042                   TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
8079  043070  001012                           BNE     64$                     ;;BRANCH IF YES
8080  043072  123727  001230  000001           CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
8081  043100  001406                           BEQ     64$                     ;;BRANCH IF YES
8082  043102  023727  001140  000176           CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
8083  043110  001005                           BNE     65$                     ;;BRANCH IF NO
8084  043112  104406                           GTSWR                           ;;GET SOFT-SWR SETTINGS
8085  043114  000403                           BR      65$
8086  043116  112737  000001  001134   64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
8087  043124                           65$:
8088  043124  000207                   1$:     RTS     PC
8089
8090                                    ;
8091                                    ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
8092                                    ;DRIVS, DRIV0-DRIV7 REGISTERS APPROPRIATELY
8093                                    ;ONLY EVEN NUMBERS ALLOWED (0,2,4,6)
8094                                    ;
8095
8096  043126  104411                   GDRVS:  RDLIN
8097  043130  012600                           MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
8098  043132  012701  177774                   MOV     #-4,R1                  ;SET UP COUNT
8099  043136  112002                   1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
8100  043140  042702  177400                   BIC     #177400,R2              ;MASK HI BYTE
8101  043144  012703  005436                   MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
8102  043150  012704  000060                   MOV     #60,R4
8103
8104  043154  020402                   2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 6?
8105  043156  001415                           BEQ     3$                      ;BRANCH IF YES
8106  043160  005723                           TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
8107  043162  005204                           INC     R4
8108  043164  020427  000067                   CMP     R4,#67
8109  043170  001371                           BNE     2$                      ;S/B 0-6 OR TERMINATOR
8110  043172  005702                           TST     R2                      ;TERMINATOR=0
8111  043174  001025                           BNE     4$
8112  043176  020127  177774                   CMP     R1,#-4
```

```
 8113  043202  001431                    BEQ     6$              ;DEFAULT ALL DRIVES
 8114  043204  005037  005462      7$:   CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
 8115  043210  000207                    RTS     PC              ;FOUND TERMINATOR, EXIT
 8116
 8117  043212  032704  000001      3$:   BIT     #BIT0,R4        ;SEE IF ODD #
 8118  043216  001014                    BNE     4$
 8119  043220  005213                    INC     @R3             ;SET UP FLAG FOR THE DRIVE
 8120  043222  005237  005434            INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
 8121  043226  112002                    MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
 8122  043230  042702  177400            BIC     #177400,R2      ;MASK
 8123  043234  022702  000054            CMP     #54,R2          ;IS IT A COMMA?
 8124  043240  001407                    BEQ     5$              ;YES, GO TO NEXT WORD.
 8125  043242  005702                    TST     R2              ;NO, IS IT A TERMINATOR?
 8126  043244  001001                    BNE     4$              ;IF NOT, SOMETHING WRONG.
 8127  043246  000756                    BR      7$              ;FOUND TERMINATOR, EXIT
 8128
 8129  043250  104401  057054      4$:   TYPE    ,EM1            ;ONLY 0,2,4,6 ALLOWED.
 8130  043254  000137  010054            JMP     PRGSRT          ;START ALL OVER
 8131
 8132  043260  005201                5$:   INC     R1              ;S/B NO MORE THAN 4 DIFF
 8133  043262  001325                    BNE     1$              ;DRIVES TYPED IN.
 8134  043264  000771                    BR      4$              ;IF NORE, HAVE ERROR.
 8135
 8136  043266  005237  005462      6$:   INC     SIZFLG          ;DO TEST 1 (SIZING)
 8137  043272  000207                    RTS     PC              ;EXIT.
 8138
 8139                                    ;
 8140                                    ;ROUTINE TO INPUT RKBAS OR DEFAULT.
 8141                                    ;
 8142
 8143  043274  104412            GBA:    RDOCT
 8144  043276  012600                    MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 8145  043300  005700                    TST     R0
 8146  043302  001403                    BEQ     1$              ;BRANCH IF DEFAULT.
 8147  043304  010037  001264            MOV     R0,$BASE
 8148  043310  000207                    RTS     PC
 8149  043312  012737  177440  001264  1$:   MOV     #177440,$BASE   ;DEFAULT VALUE
 8150  043320  000207                    RTS     PC
 8151
 8152                                    ;
 8153                                    ;ROUTINE TO INPUT RKVEC OR DEFAULT
 8154                                    ;
 8155
 8156  043322  104412            GINT:   RDOCT
 8157  043324  012600                    MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
 8158  043326  005700                    TST     R0
 8159  043330  001405                    BEQ     1$              ;BRANCH IF DEFAULT
 8160  043332  010037  001314            MOV     R0,RKVEC
 8161  043336  004737  043354      2$:   JSR     PC,SETINT
 8162  043342  000207                    RTS     PC
 8163  043344  012737  000210  001314  1$:   MOV     #210,RKVEC      ;DEFAULT VALUE
 8164  043352  000771                    BR      2$
 8165
 8166                                    ;
 8167                                    ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
 8168                                    ;
```

```
 8169
 8170  043354  013700  001314          SETINT: MOV     RKVEC,R0
 8171  043360  012720  050066                  MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
 8172  043364  013710  001316                  MOV     RKPR1,(R0)      ;PR5 TO RKVEC+2
 8173  043370  000207                          RTS     PC
 8174
 8175                                  ;
 8176                                  ;THIS ROUTINE SETS CDT IN RKCS1 IF DRIVE UNDER TEST IS AN RK07.
 8177                                  ;ENTER WITH COMMAND IN HCS1
 8178                                  ;
 8179  043372  053737  001170  005314  DOCMD:  BIS     $TMP4,HCS1      ;ADD CDT IF RK07
 8180  043400  013765  005314  000000          MOV     HCS1,RKCS1(R5)  ;DO COMMAND
 8181  043406  013737  001400  005352          MOV     T10,TEMP1
 8182  043414  004737  043466                  JSR     PC,FRDY         ;FIND CONTR READY
 8183  043420  000207                          RTS     PC              ;SET HERE IF NOT RDY
 8184  043422  062716  000002                  ADD     #2,(SP)         ;ELSE SKIP OVER ERROR
 8185  043426  000207                          RTS     PC
 8186                                  ;
 8187                                  ;THIS ROUTINE IS SIMILAR TO THE ABOVE BUT IS USED FOR DATA TRANSFERS
 8188                                  ;& REQUIRES A LONGER TIMEOUT
 8189                                  ;
 8190  043430  053737  001170  005314  DATCMD: BIS     $TMP4,HCS1      ;ADD CDT IF RK07
 8191  043436  013765  005314  000000          MOV     HCS1,RKCS1(R5)  ;DO CMD
 8192  043444  013737  001412  005352          MOV     T50000,TEMP1
 8193  043452  004737  043466                  JSR     PC,FRDY         ;FIND CONTR RDY
 8194  043456  000207                          RTS     PC
 8195  043460  062716  000002                  ADD     #2,(SP)
 8196  043464  000207                          RTS     PC
 8197
 8198                                  ;
 8199                                  ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
 8200                                  ;ENTER WITH A COUNT IN TEMP1
 8201                                  ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
 8202                                  ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
 8203                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 8204                                  ;
 8205  043466  032765  000200  000000  FRDY:   BIT     #RDY,RKCS1(R5)
 8206  043474  001010                          BNE     1$
 8207  043476  005337  005352                  DEC     TEMP1
 8208  043502  001371                          BNE     FRDY
 8209  043504  004737  043622                  JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
 8210  043510  004737  045064                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
 8211  043514  000207                          RTS     PC              ;NO RDY, EXIT
 8212  043516  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 8213  043522  004737  043622                  JSR     PC,HOLD
 8214  043526  004737  045064                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
 8215  043532  000207                          RTS     PC
 8216                                  ;
 8217                                  ;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
 8218                                  ;
 8219  043534  032765  000200  000000  FRDY1:  BIT     #RDY,RKCS1(R5)
 8220  043542  001014                          BNE     1$
 8221  043544  005337  005352                  DEC     TEMP1
 8222  043550  001371                          BNE     FRDY1
 8223  043552  016537  000034  005342          MOV     RKMR2(R5),HMR2
 8224  043560  016537  000036  005344          MOV     RKMR3(R5),HMR3
```

```
8225  043566  004737  045064                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
8226  043572  000207                          RTS     PC              ;NO RDY, EXIT
8227  043574  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
8228  043600  016537  000034  005342          MOV     RKMR2(R5),HMR2
8229  043606  016537  000036  005344          MOV     RKMR3(R5),HMR3
8230  043614  004737  045064                  JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
8231  043620  000207                          RTS     PC
8232
8233                                  ;
8234                                  ;STORE ALL RK611 REGISTERS IN HOLDING REGS
8235                                  ;
8236
8237  043622  016537  000000  005314  HOLD:   MOV     RKCS1(R5),HCS1
8238  043630  016537  000010  005316          MOV     RKCS2(R5),HCS2
8239  043636  016537  000002  005320          MOV     RKWC(R5),HWC
8240  043644  016537  000004  005322          MOV     RKBA(R5),HBA
8241  043652  016537  000006  005324          MOV     RKDA(R5),HDA
8242  043660  016537  000012  005326          MOV     RKDS(R5),HDS
8243  043666  016537  000014  005330          MOV     RKER(R5),HER
8244  043674  016537  000016  005332          MOV     RKASOF(R5),HASOF
8245  043702  016537  000020  005334          MOV     RKDC(R5),HDC
8246  043710  016537  000026  005340          MOV     RKMR1(R5),HMR1
8247  043716  016537  000034  005342          MOV     RKMR2(R5),HMR2
8248  043724  016537  000036  005344          MOV     RKMR3(R5),HMR3
8249  043732  016537  000030  005346          MOV     RKECPS(R5),HPOS
8250  043740  016537  000032  005350          MOV     RKECPT(R5),HPAT
8251  043746  000207                          RTS     PC
8252
8253                                  ;
8254                                  ;ROUTINE TO CHECK FOR CORRECT ATTN
8255                                  ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
8256                                  ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
8257                                  ;
8258  043750  010446                  TSTATN: MOV     R4,-(SP)        ;SAV R4
8259  043752  013704  001222                  MOV     $UNIT,R4
8260  043756  063704  005464                  ADD     UNITB,R4        ;ADD 1 IF ON PORT B
8261  043762  136437  005304  005333          BITB    ATTN(R4),HASOF+1
8262  043770  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
8263  043772  012604                          MOV     (SP)+,R4        ;RESTOR R4
8264  043774  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
8265  044000  000207                          RTS     PC
8266  044002  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
8267  044004  000207                          RTS     PC
8268
8269                                  ;
8270                                  ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
8271                                  ;ENTER WITH TIME IN SECONDS IN TEMP2
8272                                  ;RETURN IF NO ATTN (ERROR CONDITION)
8273                                  ;RETURN +2 IF ATTN FOUND
8274                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8275                                  ;
8276
8277  044006  010446                  FATT1:  MOV     R4,-(SP)        ;SAV R4
8278  044010  012737  177777  005352  3$:     MOV     #-1,TEMP1
8279  044016  013704  001222                  MOV     $UNIT,R4
8280  044022  063704  005464                  ADD     UNITB,R4        ;ADD 1 IF ON PORT B
```

```
8281  044026  136465  005304  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
8282  044034  001014                          BNE     2$
8283  044036  005337  005352                  DEC     TEMP1
8284  044042  001371                          BNE     1$
8285  044044  005337  005354                  DEC     TEMP2
8286  044050  001357                          BNE     3$
8287  044052  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
8288  044056  004737  045146                  JSR     PC,GSTAT        ;GET LATEST STATUS
8289  044062  012604                          MOV     (SP)+,R4        ;RESTOR R4
8290  044064  000207                          RTS     PC
8291  044066  005065  000026          2$:     CLR     RKMR1(R5)
8292  044072  004737  045146                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
8293  044076  012604                          MOV     (SP)+,R4        ;RESTOR R4
8294  044100  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
8295  044104  000207                          RTS     PC
8296
8297                                  ;
8298                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
8299                                  ;ENTER WITH COUNT IN TEMP1
8300                                  ;RETURN IF NO ATTN (ERROR)
8301                                  ;RETURN +2 IF ATTN FOUND
8302                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8303                                  ;
8304
8305  044106  010446          FATT2:  MOV     R4,-(SP)        ;SAV R4
8306  044110  013704  001222  2$:     MOV     $UNIT,R4
8307  044114  063704  005464          ADD     UNITB,R4        ;ADD 1 IF ON PORT B
8308  044120  136465  005304  000017  BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
8309  044126  001011                  BNE     1$
8310  044130  005337  005352          DEC     TEMP1
8311  044134  001365                  BNE     2$
8312  044136  005065  000026          CLR     RKMR1(R5)       ;SELECT WORD 0
8313  044142  004737  045146          JSR     PC,GSTAT        ;GET LATEST STATUS.
8314  044146  012604                  MOV     (SP)+,R4        ;RESTOR R4
8315  044150  000207                  RTS     PC
8316  044152  005065  000026  1$:     CLR     RKMR1(R5)
8317  044156  004737  045146          JSR     PC,GSTAT
8318  044162  012604                  MOV     (SP)+,R4        ;RESTOR R4
8319  044164  062716  000002          ADD     #2,(SP)         ;SKIP OVER ERROR
8320  044170  000207                  RTS     PC
8321                                  ;
8322                                  ;THIS ROUTINE CHECKS 'DRIVE AVAILABLE' IN MSG A0 AFTER A SELECT COMMAND
8323                                  ; IF NOT SET, IT DOES A NORMAL RETURN
8324                                  ; IF SET, IT DOES A RETURN+2
8325                                  ;
8326  044172  004737  045146  DRAV:   JSR     PC,GSTAT
8327  044176  032737  000040  005342  BIT     #D.DRA,HMR2     ;SEE IF DRIVE AVAILABLE SET
8328  044204  001402                  BEQ     1$              ;BR IF NO & RETURN
8329  044206  062716  000002          ADD     #2,(SP)         ;ELSE RETURN+2
8330  044212  000207          1$:     RTS     PC
8331
8332                                  ;
8333                                  ;THIS ROUTINE LOOKS FOR ATTN
8334                                  ; RETURN+2 IF FOUND BEFORE COUNT=0
8335                                  ; RETURN IF COUNT=0 (ERROR CONDITION)
8336                                  ;
```

```
8337  044214  136465  005304  000017  FATT3:  BITB    ATTN(R4),RKASOF+1(R5)  ;TEST FOR ATTN
8338  044222  001004                          BNE     1$
8339  044224  005737  001366                  TST     COUNT                  ;SEE IF TIME UP
8340  044230  001371                          BNE     FATT3                  ;BR IF NO
8341  044232  000207                          RTS     PC
8342  044234  062716  000002          1$:     ADD     #2,(SP)                ;JUMP OVER ERROR
8343  044240  000207                          RTS     PC
8344
8345                                  ;
8346                                  ;ENTER WITH A COUNT IN TEMP1
8347                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
8348                                  ;WHEN COUNT IS 0.  BASED ON AN 11/05
8349                                  ;
8350  044242  005737  005352          DLY:    TST     TEMP1                  ;5.6 US
8351  044246  001403                          BEQ     1$                     ;2.5 US
8352  044250  005337  005352                  DEC     TEMP1                  ;6.8 US
8353  044254  000772                          BR      DLY                    ;2.5 US
8354  044256  000207                  1$:     RTS     PC                     ;3.8 US
8355                                  ;
8356                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
8357                                  ;
8358
8359  044260  104401  056455          BYP:    TYPE    ,MSG14                 ;BYPASS DRIVE
8360  044264  010046                  MOV     R0,-(SP)               ;;SAVE R0 FOR TYPEOUT
8361                                                                 ;;TYPE DR#
8362  044266  104403                  TYPOS                          ;;GO TYPE--OCTAL ASCII
8363  044270     001                  .BYTE   1                      ;;TYPE 1 DIGIT(S)
8364  044271     000                  .BYTE   0                      ;;SUPPRESS LEADING ZEROS
8365  044272  000207                  RTS     PC
8366                                  ;
8367                                  ;THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
8368                                  ;
8369  044274  017637  000000  001460  CHKMSG: MOV     @(SP),CHKFLG           ;PASS MSGS TO BE TESTED
8370  044302  062716  000002                  ADD     #2,(SP)                ;BUMP RETURN ADDR TO 1ST ERROR
8371  044306  004737  045210                  JSR     PC,GSTAT1              ;GET ALL ACTUAL DRIVE & CONTR STATUS
8372
8373  044312  053737  001222  005404          BIS     $UNIT,E.A0             ;SET UNIT #
8374  044320  063737  005464  005404          ADD     UNITB,E.A0             ;ADD 1 IF ON PORT B
8375  044326  053737  001222  005410          BIS     $UNIT,E.A1
8376  044334  063737  005464  005410          ADD     UNITB,E.A1
8377  044342  053737  001222  005414          BIS     $UNIT,E.A2
8378  044350  063737  005464  005414          ADD     UNITB,E.A2
8379  044356  053737  001222  005420          BIS     $UNIT,E.A3
8380  044364  063737  005464  005420          ADD     UNITB,E.A3
8381  044372  053737  012776  005404          BIS     E.DDT,E.A0             ;SET CDT IF RK07
8382
8383  044400  013746  005352                  MOV     TEMP1,-(SP)            ;SAVE TEMP1
8384
8385  044404  013737  005404  005352          MOV     E.A0,TEMP1
8386  044412  004737  047224                  JSR     PC,SBPAR               ;GET PARITY FOR MSG A0
8387  044416  013737  005352  005404          MOV     TEMP1,E.A0
8388
8389  044424  013737  005410  005352          MOV     E.A1,TEMP1
8390  044432  004737  047224                  JSR     PC,SBPAR               ;GET PARITY FOR MSG A1
8391  044436  013737  005352  005410          MOV     TEMP1,E.A1
8392
```

```
 8393  044444  013737  005414  005352          MOV     E.A2,TEMP1
 8394  044452  004737  047224                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
 8395  044456  013737  005352  005414          MOV     TEMP1,E.A2
 8396
 8397  044464  013737  005406  005352          MOV     E.B0,TEMP1
 8398  044472  004737  047224                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
 8399  044476  013737  005352  005406          MOV     TEMP1,E.B0
 8400
 8401  044504  013737  005412  005352          MOV     E.B1,TEMP1
 8402  044512  004737  047224                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
 8403  044516  013737  005352  005412          MOV     TEMP1,E.B1
 8404
 8405  044524  013737  005416  005352          MOV     E.B2,TEMP1
 8406  044532  004737  047224                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
 8407  044536  013737  005352  005416          MOV     TEMP1,E.B2
 8408
 8409  044544  013737  005422  005352          MOV     E.B3,TEMP1
 8410  044552  004737  047224                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
 8411  044556  013737  005352  005422          MOV     TEMP1,E.B3
 8412
 8413  044564  012637  005352                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
 8414  044570  013737  001176  001172          MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE
 8415
 8416  044576  023737  005364  005404          CMP     H.A0,E.A0       ;TEST MSG A0
 8417  044604  001411                          BEQ     2$              ;BR IF OK
 8418  044606  012737  044620  001176          MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
 8419  044614  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR.
 8420  044616  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
 8421
 8422  044620  032777  001000  134312  1$:     BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
 8423  044626  001107                          BNE     20$             ;& BR IF LOOP ON ERROR
 8424  044630  062716  000002          2$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
 8425
 8426  044634  023737  005366  005406          CMP     H.B0,E.B0       ;TEST MSG B0
 8427  044642  001411                          BEQ     5$              ;BR IF OK
 8428  044644  012737  044656  001176          MOV     #4$,$ESCAPE     ;ELSE SETUP ESCAPE
 8429  044652  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR
 8430  044654  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR
 8431
 8432  044656  032777  001000  134254  4$:     BIT     #SW9,@SWR       ;RETUPN HERE FROM MAINLINE ERROR
 8433  044664  001070                          BNE     20$             ;& BR IF LOOP ON ERROR
 8434  044666  062716  000002          5$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR
 8435
 8436  044672  023737  005370  005410          CMP     H.A1,E.A1       ;TEST MSG A1
 8437  044700  001411                          BEQ     8$              ;BR IF OK
 8438  044702  012737  044714  001176          MOV     #7$,$ESCAPE
 8439  044710  011646                          MOV     (SP),-(SP)
 8440  044712  000207                          RTS     PC
 8441
 8442  044714  032777  001000  134216  7$:     BIT     #SW9,@SWR
 8443  044722  001051                          BNE     20$
 8444  044724  062716  000002          8$:     ADD     #2,(SP)
 8445
 8446  044730  023737  005372  005412          CMP     H.B1,E.B1       ;TEST MSG B1
 8447  044736  001411                          BEQ     11$             ;BR IF OK
 8448  044740  012737  044752  001176          MOV     #10$,$ESCAPE
```

```
8449  044746  011646                          MOV     (SP),-(SP)
8450  044750  000207                          RTS     PC
8451
8452  044752  032777  001000  134160  10$:    BIT     #SW9,@SWR
8453  044760  001032                          BNE     20$
8454  044762  062716  000002          11$:    ADD     #2,(SP)
8455
8456  044766  032737  000001  001460  12$:    BIT     #T.A2,CHKFLG    ;TEST MSG A2?
8457  044774  001402                          BEQ     13$             ;BR IF NO
8458  044776  004737  046026                  JSR     PC,RCYLD        ;PUT INFO CYLDIF, DO NOT CHECK
8459  045002  032737  000002  001460  13$:    BIT     #T.B2,CHKFLG    ;TEST MSG B2?
8460  045010  001402                          BEQ     14$             ;BR IF NO
8461  045012  004737  046100                  JSR     PC,RCYLA        ;PUT INFO IN CYLADD, DO NOT CHECK
8462
8463  045016  032737  000004  001460  14$:    BIT     #T.B3,CHKFLG    ;TEST MSG B3?
8464  045024  001404                          BEQ     15$
8465  045026  004737  046136                  JSR     PC,RSEC         ;PUT INFO IN SECTOR, DO NOT CHECK
8466  045032  004737  046174                  JSR     PC,RHEAD        ;PUT INFO IN HEADA, DO NOT CHECK
8467
8468  045036  013737  001172  001176  15$:    MOV     $TMP5,$ESCAPE   ;RESTORE
8469  045044  000207                          RTS     PC
8470
8471  045046  012706  001100          20$:    MOV     #STACK,SP       ;RESET STACK PTR
8472  045052  013737  001172  001176          MOV     $TMP5,$ESCAPE   ;RESTORE
8473  045060  000177  134024                  JMP     @$LPERR
8474                                  ;
8475                                  ;THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
8476                                  ;I.E.:  IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
8477                                  ;
8478  045064  005737  001456          CKCERR: TST     BYPCERR
8479  045070  001025                          BNE     4$
8480  045072  032737  100000  005314          BIT     #CERR,HCS1
8481  045100  001001                          BNE     1$              ;BR IF CERR
8482  045102  000207                          RTS     PC
8483
8484  045104  032737  004000  005314  1$:     BIT     #CTO,HCS1
8485  045112  001402                          BEQ     2$              ;BR IF NOT CTO
8486  045114  104125                          ERROR   125             ;CTO ERROR, MSG A & B INVALID
8487  045116  000207                          RTS     PC
8488
8489  045120  032737  010000  005316  2$:     BIT     #NED,HCS2
8490  045126  001401                          BEQ     3$              ;BR IF NOT NED
8491  045130  104126                          ERROR   126             ;NED ERROR, MSG A & B INVALID
8492
8493  045132  032737  001000  005316  3$:     BIT     #MDS,HCS2
8494  045140  001401                          BEQ     4$
8495  045142  104127                          ERROR   127             ;MDS ERROR, MSG A & B INVALID
8496
8497  045144  000207                  4$:     RTS     PC
8498
8499                                  ;
8500                                  ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
8501                                  ;IT THEN WAITS FOR CONTROLLER READY
8502                                  ;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
8503                                  ;
8504
```

```
 8505  045146  013746  005352          GSTAT:  MOV     TEMP1,-(SP)      ;SAVE TEMP1
 8506  045152  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 8507  045160  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
 8508  045166  012737  000001  005314          MOV     #SELDRV,HCS1
 8509  045174  004737  043372                  JSR     PC,DOCMD         ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 8510  045200  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRIVE CMD
 8511  045202  012637  005352                  MOV     (SP)+,TEMP1      ;RESTOR TEMP1.
 8512  045206  000207                          RTS     PC
 8513                                  ;
 8514                                  ;THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
 8515                                  ,& ALL CONTROLLER REGISTERS.
 8516                                  ;
 8517  045210  013746  005352          GSTAT1: MOV     TEMP1,-(SP)      ;SAVE TEMP1
 8518  045214  004737  043622                  JSR     PC,HOLD          ;GET ALL CONTR REG
 8519  045220  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
 8520  045226  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
 8521  045234  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
 8522  045242  012765  000003  000026          MOV     #3,RKMR1(R5)     ;SELECT WORD 3
 8523  045250  004737  045470                  JSR     PC,GSTAT2
 8524  045254  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRV CMD
 8525  045256  013737  005342  005400          MOV     HMR2,H.A3        ;STORE MSG A3
 8526  045264  013737  005344  005402          MOV     HMR3,H.B3        ;STORE MSG B3
 8527
 8528  045272  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8529  045300  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 8530  045306  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
 8531  045314  012765  000002  000026          MOV     #2,RKMR1(R5)     ;SELECT WORD 2
 8532  045322  004737  045470                  JSR     PC,GSTAT2
 8533  045326  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRV CMD
 8534  045330  013737  005342  005374          MOV     HMR2,H.A2        ;STORE MSG A2
 8535  045336  013737  005344  005376          MOV     HMR3,H.B2        ;STORE MSG B2
 8536
 8537  045344  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8538  045352  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 8539  045360  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
 8540  045366  012765  000001  000026          MOV     #1,RKMR1(R5)     ;SELECT WORD 1
 8541  045374  004737  045470                  JSR     PC,GSTAT2
 8542  045400  104117                          ERROR   117              ;RDY NOT SET BY END OF SELECT DRV CMD
 8543  045402  013737  005342  005370          MOV     HMR2,H.A1        ;STORE MSG A1
 8544  045410  013737  005344  005372          MOV     HMR3,H.B1        ;STORE MSG B1
 8545
 8546  045416  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 8547  045424  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
 8548  045432  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
 8549  045440  004737  045470                  JSR     PC,GSTAT2
 8550  045444  104117                          ERROR   117              ;RDY NOT SET BY END OF SEL DRV CMD
 8551  045446  013737  005342  005364          MOV     HMR2,H.A0        ;STORE MSG A0
 8552  045454  013737  005344  005366          MOV     HMR3,H.B0        ;STORE MSG B0
 8553
 8554  045462  012637  005352                  MOV     (SP)+,TEMP1      ;RESTORE TEMP1
 8555  045466  000207                          RTS     PC
 8556
 8557  045470  012737  000001  005314  GSTAT2: MOV     #SELDRV,HCS1
 8558  045476  053737  001170  005314          BIS     $TMP4,HCS1       ;ADD CDT IF RK07
 8559  045504  013765  005314  000000          MOV     HCS1,RKCS1(R5)  ;GET STATUS
 8560  045512  013737  001400  005352          MOV     T10,TEMP1
```

```
8561  045520  004737  043534                  JSR     PC,FRDY1        ;FIND CONTR RDY & STORE DRIVE REGS ONLY
8562  045524  000207                          RTS     PC              ;RET HERE IF NOT RDY
8563  045526  062716  000002                  ADD     #2,(SP)         ;RET HERE IF OK
8564  045532  000207                          RTS     PC
8565
8566                                  ;
8567                                  ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
8568                                  ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
8569                                  ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
8570                                  ;RETURN IF CERR SET
8571                                  ;RETURN +2 IF CERR CLEAR
8572                                  ;
8573  045534  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
8574  045542  013737  001400  005352          MOV     T10,TEMP1
8575  045550  004737  043466                  JSR     PC,FRDY         ;FIND RDY
8576  045554  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
8577  045556  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8578  045564  063765  005464  000010          ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
8579  045572  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
8580  045576  004737  045146                  JSR     PC,GSTAT        ;GET STATUS
8581  045602  032737  100000  005314          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
8582  045610  001401                          BEQ     1$
8583  045612  000207                          RTS     PC
8584  045614  062716  000002          1$:     ADD     #2.(SP)         ;SKIP OVER ERROR
8585  045620  000207                          RTS     PC
8586
8587                                  ;
8588                                  ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
8589                                  ;
8590  045622  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
8591  045630  004737  045146                  JSR     PC,GSTAT
8592  045634  013737  005344  001374          MOV     HMR3,SECTOR
8593  045642  042737  177017  001374          BIC     # C<M.SECT>,SECTOR
8594  045650  006237  001374                  ASR     SECTOR          ;RIGHT JUSTIFY
8595  045654  006237  001374                  ASR     SECTOR          ;SECTOR
8596  045660  006237  001374                  ASR     SECTOR          ;INFO
8597  045664  006237  001374                  ASR     SECTOR
8598  045670  000207                          RTS     PC
8599                                  ;
8600                                  ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
8601                                  ;
8602  045672  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
8603  045700  004737  045146                  JSR     PC,GSTAT
8604  045704  013737  005342  001356          MOV     HMR2,CYLDIF
8605  045712  043737  012774  001356          BIC     MASK1,CYLDIF
8606  045720  006237  001356                  ASR     CYLDIF          ;RIGHT JUSTIFY
8607  045724  006237  001356                  ASR     CYLDIF          ;CYL DIFF/OFFSET
8608  045730  006237  001356                  ASR     CYLDIF          ;INFO
8609  045734  006237  001356                  ASR     CYLDIF
8610  045740  023737  001356  012772          CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
8611  045746  001002                          BNE     1$              ;BR IF NOT
8612  045750  005037  001356                  CLR     CYLDIF          ;CLR IF YES
8613  045754  000207                  1$:     RTS     PC
8614
8615                                  ;
8616                                  ;READ THE CYL. ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
```

```
 8617                                           ;
 8618  045756  012765  000002  000026  RDCYLA:  MOV     #2,RKMR1(R5)     ;WORD 2
 8619  045764  004737  045146                   JSR     PC,GSTAT
 8620  045770  013737  005344  001360           MOV     HMR3,CYLADD
 8621  045776  043737  012774  001360           BIC     MASK1,CYLADD
 8622  046004  006237  001360                   ASR     CYLADD           ;RIGHT JUSTIFY
 8623  046010  006237  001360                   ASR     CYLADD           ;CYL ADDR
 8624  046014  006237  001360                   ASR     CYLADD           ;INFO
 8625  046020  006237  001360                   ASR     CYLADD
 8626  046024  000207                           RTS     PC
 8627                                           ;
 8628                                           ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
 8629                                           ;
 8630  046026  013737  005374  001356  RCYLD:   MOV     H.A2,CYLDIF
 8631  046034  043737  012774  001356           BIC     MASK1,CYLDIF  ;CLEAR UNWANTED INFO
 8632  046042  006237  001356                   ASR     CYLDIF          ;RIGHT JUSTIFY
 8633  046046  006237  001356                   ASR     CYLDIF
 8634  046052  006237  001356                   ASR     CYLDIF
 8635  046056  006237  001356                   ASR     CYLDIF
 8636  046062  023737  001356  012772           CMP     CYLDIF,MASK      ;CHK TO SEE IF RET IN COMPL. FORM
 8637  046070  001002                           BNE     1$               ;BR IF NO
 8638  046072  005037  001356                   CLR     CYLDIF           ;ELSE CLEAR
 8639  046076  000207                   1$:     RTS     PC
 8640                                           ;
 8641                                           ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
 8642                                           ;
 8643  046100  013737  005376  001360  RCYLA:   MOV     H.B2,CYLADD
 8644  046106  043737  012774  001360           BIC     MASK1,CYLADD  ;CLEAR UNWANTED INFO
 8645  046114  006237  001360                   ASR     CYLADD          ;RIGHT JUSTIFY
 8646  046120  006237  001360                   ASR     CYLADD
 8647  046124  006237  001360                   ASR     CYLADD
 8648  046130  006237  001360                   ASR     CYLADD
 8649  046134  000207                           RTS     PC
 8650                                           ;
 8651                                           ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
 8652                                           ;
 8653  046136  013737  005402  001374  RSEC:    MOV     H.B3,SECTOR
 8654  046144  042737  177017  001374           BIC     #C<M.SECT>,SECTOR  ,CLEAR UNWANTED INFO
 8655  046152  006237  001374                   ASR     SECTOR           ;RIGHT JUSTIFY
 8656  046156  006237  001374                   ASR     SECTOR
 8657  046162  006237  001374                   ASR     SECTOR
 8658  046166  006237  001374                   ASR     SECTOR
 8659  046172  000207                           RTS     PC
 8660                                           ;
 8661                                           ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
 8662                                           ;
 8663  046174  013737  005402  001422  RHEAD:   MOV     H.B3,HEADA
 8664  046202  042737  170777  001422           BIC     #C<M.HEAD>,HEADA  ;CLEAR UNWANTED INFO
 8665  046210  006237  001422                   ASR     HEADA            ;RIGHT JUSTIFY IT
 8666  046214  000337  001422                   SWAB    HEADA
 8667  046220  000207                           RTS     PC
 8668                                           ;
 8669                                           ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
 8670                                           ;ENTER WITH TIME IN SECONDS IN TEMP2
 8671                                           ;RETURN IF NOT FOUND
 8672                                           ;RETURN+2 IF FOUND - SKIP OVER ERROR
```

```
 8673                                           ;
 8674  046222  012737  177777  005352  FHDHM:   MOV     #-1,TEMP1       ;ALL 1'S
 8675  046230  012765  000001  000026           MOV     #1,RKMR1(R5)    ;WORD 1
 8676  046236  004737  045146          1$:      JSR     PC,GSTAT
 8677  046242  032737  000040  005342           BIT     #D.HDHM,HMR2
 8678  046250  001007                           BNE     2$
 8679  046252  005337  005352                   DEC     TEMP1
 8680  046256  001367                           BNE     1$
 8681  046260  005337  005354                   DEC     TEMP2
 8682  046264  001356                           BNE     FHDHM
 8683  046266  000207                           RTS     PC
 8684  046270  062716  000002          2$:      ADD     #2,(SP)         ;SKIP OVER ERROR
 8685  046274  000207                           RTS     PC
 8686                                           ;
 8687                                           ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
 8688                                           ;RETURN IF NOT FOUND
 8689                                           ;RETURN+2 IF FOUND:  SKIP OVER ERROR
 8690                                           ;
 8691  046276  012737  000372  005352  FLOAD:   MOV     #250.,TEMP1
 8692  046304  012765  000001  000026           MOV     #1,RKMR1(R5)    ;WORD 1
 8693  046312  004737  045146          1$:      JSR     PC,GSTAT
 8694  046316  032737  010000  005342           BIT     #D.LOAD,HMR2
 8695  046324  001004                           BNE     2$
 8696  046326  005337  005352                   DEC     TEMP1
 8697  046332  001367                           BNE     1$
 8698  046334  000207                           RTS     PC
 8699  046336  062716  000002          2$:      ADD     #2,(SP)         ;SKIP OVER ERROR
 8700  046342  000207                           RTS     PC
 8701                                           ;
 8702                                           ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
 8703                                           ;ENTER WITH CYL # IN 'CALADD'
 8704                                           ;ENTER WITH HEAD # IN 'HEAD'
 8705                                           ;ENTER WITH FORMAT IN 'FORMAT'
 8706                                           ;
 8707  046344  010046                  FHDTAB:  MOV     R0,-(SP)        ;SAV R0
 8708  046346  010146                           MOV     R1,-(SP)        ;SAV R1
 8709  046350  012700  001462                   MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
 8710  046354  005001                           CLR     R1              ;SECTOR COUNTER
 8711  046356  013737  001420  001424           MOV     HEAD,HD1
 8712  046364  006337  001424                   ASL     HD1
 8713  046370  006337  001424                   ASL     HD1
 8714  046374  006337  001424                   ASL     HD1
 8715  046400  006337  001424                   ASL     HD1
 8716  046404  006337  001424                   ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
 8717  046410  013737  001426  001430           MOV     FORMAT,FMT1
 8718  046416  000337  001430                   SWAB    FMT1
 8719  046422  006337  001430                   ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
 8720
 8721  046426  013720  001362          1$:      MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
 8722  046432  010110                           MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
 8723  046434  053710  001424                   BIS     HD1,(R0)        ;              -HEAD NO
 8724  046440  053710  001430                   BIS     FMT1,(R0)       ;              -FORMAT
```

```
 8725  046444  004737  046524                  JSR     PC,SECFLG
```

```
 8726
 8727  046450  013737  001362  005352          MOV     CALADD,TEMP1
 8728  046456  011037  005354                  MOV     (R0),TEMP2
 8729  046462  043737  001362  005354          BIC     CALADD,TEMP2
 8730  046470  042037  005352                  BIC     (R0)+,TEMP1
 8731  046474  053737  005352  005354          BIS     TEMP1,TEMP2
 8732  046502  013720  005354                  MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
 8733
 8734  046506  005201                          INC     R1              ;SECTOR CTR
 8735  046510  020127  000026                  CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
 8736  046514  001344                          BNE     1$              ;BR IF NO
 8737
 8738  046516  012601                          MOV     (SP)+,R1        ;RESTOR R1
 8739  046520  012600                          MOV     (SP)+,R0        ;RESTOR R0
 8740  046522  000207                          RTS     PC
 8741                                  ;
 8742                                  ;THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
 8743                                  ;TEST & SETS BITS 14 & 15 APPROPRIATLY.
 8744                                  ;
```

```
8745  046524  010246                  SECFLG: MOV     R2,-(SP)        ;SAVE R2
8746  046526  005737  001426                  TST     FORMAT
8747  046532  001016                          BNE     1$              ;BR IF 20 SECTOR FORMAT
8748                                  ;NOTE:  ONLY 22 SECTOR PERFORMED
8749  046534  012702  002306                  MOV     #BSE22H+8.,R2
8750  046540  004737  046574                  JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
8751  046544  052710  100000                  BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
8752
8753  046550  012702  003306                  MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
8754  046554  004737  046574                  JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
8755  046560  052710  040000                  BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
8756
8757  046564  012602                          MOV     (SP)+,R2        ;ELSE RETURN HERE
8758  046566  000207                          RTS     PC
8759
8760  046570  012602                  1$:     MOV     (SP)+,R2        ;RESTORE R2
8761  046572  000207                          RTS     PC
8762
8763                                  ;
8764                                  ;THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
8765                                  ;ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
8766                                  ;RETURN IF NO COMPARE
8767                                  ;RETURN+4 IF COMPARE
8768                                  ;
8769  046574  010346                  FLGTST: MOV     R3,-(SP)        ;SAVE R3
8770
8771  046576  021227  177777          1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
8772  046602  001002                          BNE     2$              ;BR IF NO
8773  046604  012603                          MOV     (SP)+,R3        ;RESTORE R3
8774  046606  000207                          RTS     PC
8775
8776  046610  022237  001362          2$:     CMP     (R2)+,CALADD    ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
8777  046614  001403                          BEQ     3$
8778  046616  062702  000002                  ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
8779  046622  000765                          BR      1$
8780
8781  046624  013703  001420          3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
8782  046630  000303                          SWAB    R3
8783  046632  050103                          BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
8784  046634  022203                          CMP     (R2)+,R3        ;SEE IF SECTOR/HEAD COMPARE
8785                                                                  ;& INCR PTR TO NEXT CYL WORD
8786  046636  001401                          BEQ     4$              ;BR IF COMPARE
8787  046640  000756                          BR      1$              ;ELSE TRY NEXT CYL
8788
8789  046642  012603                  4$:     MOV     (SP)+,R3        ;RESTORE R3
8790  046644  062716  000004                  ADD     #4,(SP)         ;INCREMENT RET ADDR
8791  046650  000207                          RTS     PC
8792
8793                                  ;
8794                                  ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
8795                                  ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
8796                                  ;
8797  046652  010046                  SORT:   MOV     R0,-(SP)        ;SAVE R0
8798  046654  010146                          MOV     R1,-(SP)        ;SAVE R1
8799  046656  004737  045622                  JSR     PC,RDSEC
8800  046662  062737  000001  001374          ADD     #1,SECTOR
```

```
 8801  046670  004737  046760                  JSR     PC,MULT6        ;MULT SECTOR BY 6
 8802
 8803  046674  012700  000204                  MOV     #132.,R0
 8804  046700  163700  001374                  SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
 8805  046704  010037  001374                  MOV     R0,SECTOR
 8806  046710  062737  001666  001374          ADD     #RHTAB,SECTOR   ;SAVE INDEX
 8807
 8808  046716  062700  001666                  ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
 8809  046722  012701  002072                  MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
 8810
 8811  046726  012021                  1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
 8812  046730  020027  002072                  CMP     R0,#RHTAB+132.
 8813  046734  001374                          BNE     1$
 8814
 8815  046736  012700  001666                  MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
 8816  046742  012021                  2$:     MOV     (R0)+,(R1)+
 8817  046744  020037  001374                  CMP     R0,SECTOR
 8818  046750  001374                          BNE     2$
 8819
 8820  046752  012601                          MOV     (SP)+,R1        ;RESTOR R1
 8821  046754  012600                          MOV     (SP)+,R0        ;RESTOR R0
 8822  046756  000207                          RTS     PC
 8823
 8824                                  ;
 8825                                  ;MULT BY 6.  ENTER WITH DESIRED # IN 'SECTOR'
 8826                                  ;
 8827  046760  006337  001374          MULT6:  ASL     SECTOR          ;2 X SECTOR
 8828  046764  013746  001374                  MOV     SECTOR,-(SP)
 8829  046770  006337  001374                  ASL     SECTOR          ;4 X SECTOR
 8830  046774  062637  001374                  ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
 8831  047000  000207                          RTS     PC
 8832
 8833                                  ;
 8834                                  ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
 8835                                  ;CMD.  IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
 8836                                  ;CYLINDER AT CYL 410, TRACK 2.
 8837                                  ;
 8838                                  ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
 8839                                  ;RETURN+2 IF LISTED, SKIP OVER ERROR
 8840                                  ;
 8841  047002  010446                  TRUERR: MOV     R4,-(SP)                ;SAVE R4
 8842
 8843  047004  032737  010000  005314          BIT     #CFMT,HCS1      ;CHECK FORMAT
 8844  047012  001014                          BNE     3$              ;BR FOR 20 SECTOR FORMAT
 8845                                                                  ;ONLY 22 SECTOR FMT IN THIS PGM
 8846
 8847  047014  012704  002306                  MOV     #BSE22H+8.,R4
 8848  047020  004737  047054                  JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
 8849  047024  000407                          BR      3$              ;RETURN HERE IF YES
 8850
 8851  047026  012704  003306                  MOV     #BSE22S+8.,R4   ;ELSE RETURN HERE
 8852  047032  004737  047054                  JSR     PC,TERR1        ;& SEE IF ON SOFTWARE DETECTED TABLE
 8853  047036  000402                          BR      3$              ;RETURN HERE IF YES
 8854
 8855  047040  012604                  1$.     MOV     (SP)+,R4        ;RESTORE R4
 8856  047042  000207                          RTS     PC              ;RETURN WITHOUT JUMPING OVER ERROR
```

```
8857
8858  047044  012604                  3$:     MOV     (SP)+,R4        ;RESTORE R4
8859  047046  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR ON RETURN
8860  047052  000207                          RTS     PC
8861
8862                                  ;
8863                                  ;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
8864                                  ;THE BSE TABLE FOR THE ABOVE SUBROUTINE.
8865                                  ;RETURN IF FOUND ON TABLE
8866                                  ;RETURN+2 IF NOT FOUND
8867                                  ;
8868  047054  021427  177777          TERR1:  CMP     (R4),#-1        ;SEE IF ALL 1'S
8869  047060  001405                          BEQ     1$              ;BR IF YES, NOT ON TABLE
8870  047062  022437  005334                  CMP     (R4)+,HDC       ;SEE IF CYL MATCH
8871  047066  001405                          BEQ     2$              ;BR IF YES
8872  047070  005724                          TST     (R4)+           ;ELSE ADV TO NEXT CYL WORD
8873  047072  000770                          BR      TERR1           ;& TRY AGAIN.
8874
8875  047074  062716  000002          1$:     ADD     #2,(SP)
8876  047100  000207                          RTS     PC
8877
8878  047102  022437  005324          2$:     CMP     (R4)+,HDA       ;SEE IF SECTOR & TRACK MATCH
8879  047106  001401                          BEQ     3$              ;BR IF YES
8880  047110  000761                          BR      TERR1           ;OR TRY AGAIN
8881
8882  047112  000207                  3$:     RTS     PC
8883
8884
8885                                  ;
8886                                  ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
8887                                  ;
8888  047114  005037  001372          CLKON:  CLR     TIMUP
8889  047120  005737  005460                  TST     PCLKF
8890  047124  001004                          BNE     1$              ;BRANCH IF P-CLOCK PRESENT
8891  047126  012777  000100  132172        MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
8892  047134  000207                          RTS     PC
8893  047136  012777  177777  132156  1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
8894  047144  012777  000135  132146        MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
8895  047152  000207                          RTS     PC              ;LINE FREQ & RUN
8896                                  ;
8897                                  ;KW11-L & KW11-P INTERRUPT HANDLER
8898                                  ;
8899  047154  005337  001366          CLOCK:  DEC     COUNT
8900  047160  000002                          RTI
8901                                  ;
8902                                  ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
8903                                  ;
8904  047162  005737  005460          CLKOF:  TST     PCLKF
8905  047166  001003                          BNE     1$              ;BRACH IF P-CLOCK PRESENT
8906  047170  005077  132132                  CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
8907  047174  000207                          RTS     PC
8908  047176  005077  132116          1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
8909  047202  000207                          RTS     PC
8910
8911                                  ;
8912                                  ;THIS ROUTINE DOES A TIMEOUT DEPENDING ON THE VALUE IN 'COUNT' WHICH THE
```

```
8913                                    ;'CLOCK' HANDLER DECREMENTS EVERY 1/60 SEC.
8914                                    ;
8915  047204  004737  047114            TMO:    JSR     PC,CLKON        ;TURN CLOCK ON
8916  047210  005737  001366            1$:     TST     COUNT           ;TIME UP?
8917                                    ;       BNE     TMO             ;BR IF NO  S9-SEP-77
8918  047214  001375                            BNE     1$              ;29-SEP-77
8919  047216  004737  047162                    JSR     PC,CLKOF        ;ELSE, TURN CLOCK OFF
8920  047222  000207                            RTS     PC
8921
8922                                    ;
8923                                    ;THIS ROUTINE GENERATES PARITY FOR EXPECTED MESSAGES
8924                                    ;ENTER WITH THE EXPECTED WORD IN TEMP1
8925                                    ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
8926                                    ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
8927                                    ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
8928                                    ; THE PARITY BIT IS NOT SET IN B .
8929                                    ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
8930                                    ;SET IN TEMP1
8931                                    ;
8932  047224  010046                    SBPAR:  MOV     R0,-(SP)        ;SAVE R0
8933  047226  010146                            MOV     R1,-(SP)        ;SAVE R1
8934  047230  012700  000021                    MOV     #17.,R0         ;SHIFT COUNTER
8935  047234  005001                            CLR     R1              ;COUNT # OF 1'S IN TEMP1
8936  047236  000241                            CLC                     ;CLEAR CARRY
8937
8938  047240  006137  005352            1$:     ROL     TEMP1
8939  047244  103001                            BCC     2$              ;BR IF CARRY CLEAR
8940  047246  005201                            INC     R1              ;COUNT # OF 1'S
8941  047250  005300                    2$:     DEC     R0              ;SHIFT COUNTER
8942  047252  001372                            BNE     1$
8943
8944  047254  032701  000001                    BIT     #BIT0,R1
8945  047260  001003                            BNE     3$              ;BR IF ODD # IN R0
8946  047262  052737  100000  005352            BIS     #M.PAR,TEMP1    ;SET PARITY BIT
8947  047270  012601                    3$:     MOV     (SP)+,R1        ;RESTORE R1
8948  047272  012600                            MOV     (SP)+,R0        ;RESTORE R0
8949  047274  000207                            RTS     PC
8950
8951                                    ;
8952                                    ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
8953                                    ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
8954                                    ; IE: MY LOOP MACRO
8955                                    ;
8956  047276  032777  001000  131634    SCOP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
8957  047304  001406                            BEQ     1$              ;BR IF NO
8958  047306  105737  001103                    TSTB    $ERFLG          ;HAD ERROR?
8959  047312  001403                            BEQ     1$              ;BR IF NO
8960  047314  013716  001110                    MOV     $LPERR,(SP)
8961  047320  000002                            RTI
8962
8963  047322  011637  001110            1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
8964  047326  000002                            RTI
8965
8966
8967
8968
```

```
                                        ;
                                        ;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
                                        ;IT IS USED TO ALLOW THE OPERATOR TO HLAT THE CPU WHILE INSURING
                                        ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
                                        ;THE CPU.
                                        ;
047330  022626              STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT

047332  004737  045534              JSR     PC,SUBCLR
047336  104024                      ERROR   24              ;CERR AFTER

047340  005737  005276              TST     UNLD            ;SEE IF HEADS UNLOADED
047344  001431                      BEQ     3$              ;BR IF NO
047346  005737  000042              TST     42              ;SEE IF MANUAL OR AUTO MODE
047352  001403                      BEQ     1$              ;BR IF MANUAL MODE
047354  104401  056712              TYPE    ,MSG74          ;PGM ABORT PENDING
047360  000402                      BR      2$
047362  104401  056760      1$:     TYPE    ,MSG75          ;HALT PENDING
047366                      2$:

047366  004737  045534              JSR     PC,SUBCLR
047372  104024                      ERROR   24              ;CERR AFTER SCLR

047374  012737  000011  005314      MOV     #SRTSPL,HCS1
047402  004737  043372              JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
047406  104121                      ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.

047410  013737  001406  005354      MOV     T100,TEMP2      ;SETUP TIMEOUT
047416  004737  044006              JSR     PC,FATT1        ;FIND ATTN
047422  104074                      ERROR   74              ;NO ATTN AFTER ST SPIN CMD.

047424  005037  005276              CLR     UNLD

047430  005737  005300      3$:     TST     BADHDR          ;SEE IF HEADERS VALID
047434  001520                      BEQ     4$              ;BR IF YES
047436  005237  005302              INC     HPEND

047442  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
047450  013765  001222  000010      MOV     $UNIT,RKCS2(R5)
047456  063765  005464  000010      ADD     UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
047464  012737  000013  005314      MOV     #RECAL,HCS1
047472  004737  043372              JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
047476  104124                      ERROR   124             ;RDY NOT SET AFTER RECAL CMD

047500  012765  000001  000026      MOV     #1,RKMR1(R5)    ;SELECT WORD 1
047506  004737  045146              JSR     PC,GSTAT
047512  032737  020000  005342      BIT     #D.RTZ,HMR2
047520  001001                      BNE     64$
047522  104070                      ERROR   70              ;RTZ NOT SET DURING RECAL CMD
047524  013737  001400  005354  64$:MOV     T10,TEMP2       ;SETUP TIMEOUT
047532  004737  044006              JSR     PC,FATT1        ;FIND ATTN
047536  104055                      ERROR   55              ;NO ATTN AFTER RECAL CMD

047540  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
047546  013765  001222  000010      MOV     $UNIT,RKCS2(R5) ;DRIVE#
```

```
 9025  047554  063765  005464  000010          ADD     UNITB,RKCS2(R5)  ;ADD 1 IF ON PORT B
 9026  047562  012737  000005  005314          MOV     #CLEAR,HCS1
 9027  047570  004737  043372                  JSR     PC,DOCMD         ;DO DRIVE CLEAR CMD & GET CONTR RDY
 9028  047574  104151                          ERROR   151              ;NO RDY AFTER DRIVE CLEAR CMD
 9029  047576  004737  043750                  JSR     PC,TSTATN        ;TEST FOR ATTN
 9030  047602  000401                          BR      65$
 9031  047604  104154                          ERROR   154              ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 9032  047606                          65$:
 9033
 9034  047606  012737  010340  005404          MOV     #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0  ;EXPECTED MSG A0
 9035  047614  005037  005406                  CLR     E.B0             ;EXPECTED MSG B0
 9036  047620  012737  001720  005410          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 9037  047626  012737  000001  005412          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
 9038  047634  005037  005414                  CLR     E.A2             ;EXPECTED MSG A2
 9039  047640  012737  000002  005416          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 9040  047646  012737  000003  005422          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 9041
 9042  047654  004737  044274                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 9043  047660  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
 9044  047662  104033                          ERROR   33               ;MSG A0 ERROR AFTER DRV CLEAR CMD
 9045  047664  104034                          ERROR   34               ;MSH B0 ERROR
 9046  047666  104035                          ERROR   35               ;MSG A1 ERROR
 9047  047670  104036                          ERROR   36               ;MSG B1 ERROR
 9048
 9049
 9050  047672  000137  047736                  JMP     FORM             ;WRITE VALID FORMATS
 9051
 9052  047676  005737  000042          4$:     TST     42               ;SEE IF MANUAL OR AUTO MODE
 9053  047702  001410                          BEQ     5$               ;BR IF MANUAL MODE
 9054  047704  104401  057015                  TYPE    ,MSG76           ;PGM ABORTED
 9055  047710  005037  042732                  CLR     $EOPCT           ;SET UP EOP TO EXIT TO MONITOR
 9056  047714  005037  001176                  CLR     $ESCAPE
 9057  047720  000137  042704                  JMP     $EOP1            ;ABORT PGM
 9058
 9059  047724  104401  057037          5$:     TYPE    ,MSG77           ;CPU HALTED
 9060  047730  000000                          HALT
 9061  047732  000137  010646                  JMP     ST5              ;START OVER IF CONTINUE PRESSED
 9062  047736                          FORM:
 9063                                  .SBTTL  UNEXPECTED TIMEOUT HANDLER
 9064
 9065                                  ;
 9066                                  ;THIS ROUTINE IS ENTERED IF THERE IS
 9067                                  ;       A.  NON EXISTANT MEMORY (NO SSYN)
 9068                                  ;       B.  BOUNDRY ERROR
 9069                                  ;       C.  STACK OVERFLOW
 9070                                  ;
 9071
 9072  047736  011600                  BADTMO: MOV     (SP),R0          ;SAVE PC WHERE TIMEOUT OCCURRED
 9073  047740  005740                          TST     -(R0)            ;GET PC BEFORE UPDATE
 9074  047742  032777  020000  131170          BIT     #SW13,@SWR       ;INHIBIT ERR TYPOUT?
 9075  047750  001005                          BNE     1$               ;YES, DON'T TYPE
 9076  047752  104401  057217                  TYPE    ,EM3             ;ABORT TESTS,UNEXP T.O.  @ PC=
 9077  047756  010046                          MOV     R0,-(SP)         ;;SAVE R0 FOR TYPEOUT
 9078                                                                   ;;TYPE PC
 9079  047760  104403                          TYPOS                    ;;GO TYPE--OCTAL ASCII
 9080  047762     006                          .BYTE   6                ;;TYPE 6 DIGIT(S)
```

```
9081  047763  000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
9082  047764  032777  001000  131146  1$:  BIT    #SW9,@SWR       ;LOOP ON ERROR?
9083  047772  001403                       BEQ    2$              ;NO, BRANCH
9084  047774  022626                       CMP    (SP)+,(SP)+     ;YES, RESTORE STACK
9085  047776  000177  131104               JMP    @$LPADR         ;GO TO STARTING ADDR OF TEST
9086                                                              ;THAT GAVE BAD TIMEOUT
9087  050002  032777  040000  131130  2$:  BIT    #SW14,@SWR      ;LOOP ON TEST?
9088  050010  001401                       BEQ    3$              ;NO BRANCH
9089  050012  000002                       RTI                    ;YES
9090
9091  050014  000000                  3$:  HALT                   ;UNEXPECTED TIME OUT OCCURRED
9092                                                              ;AS INDICATED.  YOU CAN LOOP ON
9093                                                              ;ERROR, LOOP ON TEST OR INHIBIT
9094                                                              ;ERROR TYPEOUT BY SETTING THOSE
9095                                                              ;SWITCHES.
9096                                                              ;
9097  050016  022626                       CMP    (SP)+,(SP)+     ;RESTORE STACK
9098  050020  000137  042704               JMP    $EOP1           ;ABORT TESTS
9099
9100                          .SBTTL  MEMORY CHECK ENABLE TRAP
9101
9102  050024  012737  050040  001176  MEMERR: MOV    #1$,$ESCAPE
9103  050032  011637  001334                MOV    (SP),TRAPPC     ;STORE PC
9104  050036  104017                        ERROR  17              ;UNEXP MEM PARITY TRAP
9105  050040  005037  001176           1$:  CLR    $ESCAPE
9106  050044  032777  001000  131066        BIT    #SW9,@SWR       ;CHECK IF LOOP ON ERROR
9107  050052  001001                        BNE    2$              ;YES, FORCE STACK AND TRY AGAIN
9108  050054  000002                        RTI                    ;ELSE RETURN
9109
9110  050056  012706  001100           2$:  MOV    #STACK,SP       ;INIT STACK
9111  050062  000177  131022                JMP    @$LPERR         ;LOOP ON ERROR
9112
9113
9114                          .SBTTL  RK06 INTERRUPT HANDLER
9115
9116  050066  000240                  INTER:  NOP
9117  050070  000240                          NOP
9118  050072  000240                          NOP
9119  050074  011600                          MOV    (SP),R0         ;SAVE PC WHERE INT OCCURRED.
9120  050076  005740                          TST    -(R0)           ;GET PC BEFORE UPDATE.
9121  050100  104401  056167                 TYPE   ,MSG6           ;INT AT PC=
9122  050104  010046                          MOV    R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
9123                                                                 ;;TYPE PC
9124  050106  104403                          TYPOS                  ;;GO TYPE--OCTAL ASCII
9125  050110     006                           .BYTE  6              ;;TYPE 6 DIGIT(S)
9126  050111     000                          .BYTE   0              ;;SUPPRESS LEADING ZEROS
9127  050112  000000                          HALT
9128  050114  000240                          NOP
9129  050116  000240                          NOP
9130  050120  000002                          RTI
9131
9132                          .SBTTL  POWER DOWN AND UP ROUTINES
9133
9134                          ;POWER DOWN ROUTINE
9135
9136  050122  012737  050134  000024  $PWRDN: MOV    #$PWRUP,PWRVEC  ;SET UP VECTOR
```

```
                                        HALT
                                        BR      .-2             ;HANG UP.

                                ;POWER UP ROUTINE

                                $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
                                1$:     INC     $PWRCT          ;WAIT FOR THE INCR
                                        BNE     1$              ;OF WORD
                                        MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
                                        MOV     #PR7,PWRVEC+2   ;PRIORITY 7
                                        MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
                                        MOV     #STACK,SP       ;INITIALIZE STACK
                                        TYPE    ,MSG11          ;REPORT POWER FAIL
                                        RESET
                                        JMP     PFSRT

                                $PWRCT: 0                       ;WAIT COUNT FOR TTY

                                ;
                                ;DIVISION UTILITY ROUTINE
                                ;
                                ;R0-R1-R2-R3=DIVIDEND
                                ;R4-R5=DIVISOR
                                ;R0-R1=REMAINDER AFTER DIVISION
                                ;R2-R3=QUOTIENT AFTER DIVISION
                                ;ENTER WITH JSR PC,M.DPID
                                ;
                                M.DPID: MOV     #40,-(SP)       ;COUNTER FOR DIVISION CYCLES
                                        MOV     R4,-(SP)        ;HI ORDER
                                        MOV     R5,-(SP)        ;LO ORDER TO THE STACK
                                        NEG     2(SP)           ;FORM NEGATIVE
                                        NEG     @SP             ;VERSION OF DIVISOR
                                        SBC     2(SP)
                                        ADD     @SP,R1
                                        ADC     R0              ;PERFORM INIT SUBT.
                                        ADD     2(SP),R0
                                        BCS     M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
                                        CLR     -(SP)           ;THIS IS A LONGER LASTING CARRY BIT
                                M.DP40: ROL     R3
                                        ROL     R2
                                        ROL     R1
                                        ROL     R0
                                        TST     @SP             ;TEST CARRY INDICATOR
                                        BEQ     M.DP41          ;IF TO CARRY THEN ADD, ELSE SUBT.
                                        CLR     @SP             ;CLEAR UP FOR NEXT TIME
                                        ADD     2(SP),R1
                                        ADC     R0              ;ADD -(DIVISOR)
                                        ADC     @SP             ;SET CARRY
                                        ADD     4(SP),R0
                                        BR      M.DP42

                                M.DP41: ADD     R5,R1
                                        ADC     R0              ;ADD +(DIVISOR)
                                        ADC     @SP             ;SET CARRY
                                        ADD     R4,R0
                                M.DP42: ADC     @SP             ;SET CARRY
```

```
 9193  050314  005716                        TST     @SP             ;TEST THE UPDATE INDICATOR
 9194  050316  001401                        BEQ     .+4             ;IF 0,FORGET IT
 9195  050320  005203                        INC     R3              ;NO CARRY POSSIBLE HERE
 9196  050322  005366  000006                DEC     6(SP)           ;DECREMENT CTR
 9197  050326  003347                        BGT     M.DP40          ;BR IF MORE TO DO
 9198  050330  006003                        ROR     R3
 9199  050332  103404                        BCS     M.DP44
 9200  050334  060501                        ADD     R5,R1
 9201  050336  005500                        ADC     R0
 9202  050340  060400                        ADD     R4,R0
 9203  050342  000241                        CLC
 9204
 9205  050344  006103                M.DP44: ROL     R3
 9206  050346  062706  000010                ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
 9207  050352  000242                        CLV
 9208  050354  000207                        RTS     PC
 9209
 9210  050356  062706  000006        M.DP50: ADD     #6,SP
 9211  050362  000262                        SEV
 9212  050364  000207                        RTS     PC
 9213
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS.  IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG. (DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ,*SW08=1         LOOP ON TEST IN SWR<7:0>
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

050366                          $SCOPE:
050366  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
050370  032777  040000  130542  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
050376  001114                         BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
050400  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
050402  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
050406  012737  050426  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
050414  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
050420  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050424  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
050426  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
050430  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
050434  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
050436                          6$: ;#####END OF CODE FOR THE XOR TESTER#####
050436  032777  000400  130474         BIT     #BIT08,@SWR     ;;LOOP ON SPEC.  TEST?
050444  001404                         BEQ     2$              ;;BR IF NO
050446  127737  130466  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
050454  001465                         BEQ     $OVER           ;;BR IF YES
050456  105737  001103         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
050462  001421                         BEQ     3$              ;;BR IF NO
050464  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX.  ERRORS FOR THIS TEST OCCURRED?
050472  101015                         BHI     3$              ;;BR IF NO
050474  032777  001000  130436         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
050502  001404                         BEQ     4$              ;;BR IF NO
050504  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
050512  000446                         BR      $OVER
050514  105037  001103         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
050520  005037  001174                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
050524  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
050526  032777  004000  130404  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
050534  001011                         BNE     1$              ;;BR IF YES
050536  005737  001216                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
050542  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
050544  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
050550  023737  001174  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
050556  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
050560  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
050566  013737  050644  001174         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
050574  105237  001102         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
050600  113737  001102  001214         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
9270  050606  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
9271  050612  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
9272  050616  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
9273  050622  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9274  050630  013777  001102  130304  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
9275  050636  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
9276  050642  000002                          RTI                     ;;FIXES PS
9277  050644  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
9278                                  .SBTTL  ERROR HANDLER ROUTINE
9279
9280                                  ;;**************************************************************
9281                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9282                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9283                                  ;*AND GO TO TYPERR ON ERROR
9284                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9285                                  ;*SW15=1         HALT ON ERROR
9286                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
9287                                  ;*SW10=1         BELL ON ERROR
9288                                  ;*SW09=1         LOOP ON ERROR
9289                                  ;*CALL
9290                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9291
9292  050646                  $ERROR:
9293  050646  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
9294  050650  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
9295  050654  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
9296  050656  013777  001102  130256          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
9297  050664  032777  002000  130246          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
9298  050672  001402                          BEQ     1$              ;;NO - SKIP
9299  050674  104401  001200                  TYPE    ,$BELL          ;;RING BELL
9300  050700  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
9301  050704  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
9302  050710  162737  000002  001116          SUB     #2,$ERRPC
9303  050716  117737  130174  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
9304  050724  032777  020000  130206          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
9305  050732  001004                          BNE     20$             ;;SKIP TYPEOUTS
9306  050734  004737  066736                  JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
9307  050740  104401  001205                  TYPE    ,$CRLF
9308  050744                          20$:
9309  050744  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
9310  050752  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
9311  050754  113737  001114  050766          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
9312  050762  004737  051572                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
9313  050766     000          21$:    .BYTE   0
9314  050767     000                  .BYTE   0
9315  050770  000777          22$:    BR      22$             ;;APT ERROR LOOP
9316  050772  005777  130142          2$:     TST     @SWR            ;;HALT ON ERROR
9317  050776  100002                          BPL     3$              ;;SKIP IF CONTINUE
9318  051000  000000                          HALT                    ;;HALT ON ERROR!
9319  051002  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
9320  051004  032777  001000  130126  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
9321  051012  001402                          BEQ     4$              ;;BR IF NO
9322  051014  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
9323  051020  005737  001176          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
9324  051024  001402                          BEQ     5$              ;;BR IF NONE
9325  051026  013716  001176                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
 9326  051032                          5$:
 9327  051032  022737  042772  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 9328  051040  001001                          BNE     6$              ;;BRANCH IF NO
 9329  051042  000000                          HALT                    ;;YES
 9330  051044                          6$:
 9331  051044  000002                          RTI                     ;;RETURN
 9332                                  .SBTTL  TYPE ROUTINE
 9333
 9334                                  ;;**********************************************************************
 9335                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 9336                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 9337                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 9338                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 9339                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 9340                                  ;*
 9341                                  ;*CALL:
 9342                                  ;*1) USING A TRAP INSTRUCTION
 9343                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 9344                                  ;*OR
 9345                                  ;*      TYPE
 9346                                  ;*      MESADR
 9347                                  ;*
 9348
 9349  051046  105737  001157          $TYPE:  TSTB    $TTPFLG         ;;IS THERE A TERMINAL?
 9350  051052  100002                          BPL     1$              ;;BR IF YES
 9351  051054  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 9352  051056  000430                          BR      3$              ;;LEAVE
 9353  051060  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 9354  051062  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 9355  051066  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9356  051074  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 9357  051076  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 9358  051104  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 9359  051106  010037  051116                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 9360  051112  004737  051562                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 9361  051116  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
 9362  051120  132737  000040  001231  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 9363  051126  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
 9364  051130  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 9365  051132  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 9366  051134  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 9367  051136  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 9368  051140  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 9369  051144  000002                          RTI                     ;;RETURN
 9370  051146  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 9371  051152  001430                          BEQ     8$
 9372  051154  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 9373  051160  001006                          BNE     5$
 9374  051162  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 9375  051164  104401                          TYPE                    ;;TYPE A CR AND LF
 9376  051166  001205                          $CRLF
 9377  051170  105037  051324                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 9378  051174  000755                          BR      2$              ;;GET NEXT CHARACTER
 9379  051176  004737  051260          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 9380  051202  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 9381  051206  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
 9382  051210  013746  001154                    MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 9383                                                                      ;;AND THE NULL CHAR.
 9384  051214  105366  000001          7$:       DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 9385  051220  002770                             BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 9386  051222  004737  051260                    JSR     PC,$TYPEC       ;;GO TYPE A NULL
 9387  051226  105337  051324                    DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 9388  051232  000770                             BR      7$              ;;LOOP
 9389
 9390                                    ;HORIZONTAL TAB PROCESSOR
 9391
 9392  051234  112716  000040          8$:       MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 9393  051240  004737  051260          9$:       JSR     PC,$TYPEC       ;;TYPE A SPACE
 9394  051244  132737  000007  051324            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 9395  051252  001372                             BNE     9$              ;;TAB STOP
 9396  051254  005726                             TST     (SP)+           ;;POP SPACE OFF STACK
 9397  051256  000724                             BR      2$              ;;GET NEXT CHARACTER
 9398  051260  105777  127664          $TYPEC:   TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 9399  051264  100375                             BPL     $TYPEC
 9400  051266  116677  000002  127656            MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 9401  051274  122766  000015  000002            CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 9402  051302  001003                             BNE     1$              ;;BRANCH IF NO
 9403  051304  105037  051324                    CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 9404  051310  000406                             BR      $TYPEX          ;;EXIT
 9405  051312  122766  000012  000002  1$:       CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 9406  051320  001402                             BEQ     $TYPEX          ;;BRANCH IF YES
 9407  051322  105227                             INCB    (PC)+           ;;COUNT THE CHARACTER
 9408  051324  000000                    $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 9409  051326  000207                    $TYPEX:  RTS     PC
 9410
 9411                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 9412
 9413                                    ;;**************************************************************
 9414                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 9415                                    ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT.  DEPENDING ON WHETHER THE
 9416                                    ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 9417                                    ;*BEFORE THE FIRST DIGIT OF THE NUMBER.  LEADING ZEROS WILL ALWAYS BE
 9418                                    ;*REPLACED WITH SPACES.
 9419                                    ;*CALL:
 9420                                    ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 9421                                    ;*      TYPDS                   ;;GO TO THE ROUTINE
 9422
 9423  051330                            $TYPDS:
 9424  051330  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9425  051332  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9426  051334  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9427  051336  010346                             MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9428  051340  010546                             MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9429  051342  012746  020200                     MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 9430  051346  016605  000020                     MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 9431  051352  100004                             BPL     1$              ;;BR IF INPUT IS POS.
 9432  051354  005405                             NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 9433  051356  112766  000055  000001            MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG
 9434  051364  005000                    1$:      CLR     R0              ;;ZERO THE CONSTANTS INDEX
 9435  051366  012703  051544                     MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 9436  051372  112723  000040                     MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 9437  051376  005002                    2$:      CLR     R2              ;;CLEAR THE BCD NUMBER
```

```
 9438  051400  016001  051534                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 9439  051404  160105                  3$:    SUB     R1,R5           ;;FORM THIS BCD DIGIT
 9440  051406  002402                         BLT     4$              ;;BR IF DONE
 9441  051410  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 9442  051412  000774                         BR      3$
 9443  051414  060105                  4$:    ADD     R1,R5           ;;ADD BACK THE CONSTANT
 9444  051416  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
 9445  051420  001002                         BNE     5$              ;;FALL THROUGH IF 0
 9446  051422  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 9447  051424  100407                         BMI     7$              ;;BR IF YES
 9448  051426  106316                  5$:    ASLB    (SP)            ;;MSD?
 9449  051430  103003                         BCC     6$              ;;BR IF NO
 9450  051432  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 9451  051440  052702  000060          6$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 9452  051444  052702  000040          7$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 9453  051450  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 9454  051452  005720                         TST     (R0)+           ;;JUST INCREMENTING
 9455  051454  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
 9456  051460  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
 9457  051462  003002                         BGT     8$              ;;GO TO EXIT
 9458  051464  010502                         MOV     R5,R2           ;;GET THE LSD
 9459  051466  000764                         BR      6$              ;;GO CHANGE TO ASCII
 9460  051470  105726                  8$:    TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 9461  051472  100003                         BPL     9$              ;;BR IF NO
 9462  051474  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 9463  051502  105013                  9$:    CLRB    (R3)            ;;SET THE TERMINATOR
 9464  051504  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
 9465  051506  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
 9466  051510  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
 9467  051512  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 9468  051514  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 9469  051516  104401  051544                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 9470  051522  016666  000002  000004         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 9471  051530  012616                         MOV     (SP)+,(SP)
 9472  051532  000002                         RTI                     ;;RETURN TO USER
 9473  051534  023420                  $DTBL: 10000.
 9474  051536  001750                         1000.
 9475  051540  000144                         100.
 9476  051542  000012                         10.
 9477  051544  000004                  $DBLK: .BLKW   4
 9478                                   .SBTTL  APT COMMUNICATIONS ROUTINE
 9479
 9480                                   ;;**********************************************************************
 9481  051554  112737  000001  052020  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 9482  051562  112737  000001  052016  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 9483  051570  000403                         BR      $ATYC
 9484  051572  112737  000001  052020  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 9485  051600                         $ATYC:
 9486  051600  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9487  051602  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9488  051604  105737  052016                 TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 9489  051610  001450                         BEQ     5$              ;;IF NOT:  BR
 9490  051612  122737  000001  001230         CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 9491  051620  001031                         BNE     3$              ;;IF NOT:  BR
 9492  051622  132737  000100  001231         BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 9493  051630  001425                         BEQ     3$              ;;IF NOT:  BR
```

```
 9494  051632  017600  000004                 MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 9495  051636  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9496  051644  005737  001210          1$:    TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 9497  051650  001375                         BNE     1$              ;;IF NOT:  WAIT
 9498  051652  010037  001224                 MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 9499  051656  105720                  2$:    TSTB    (R0)+           ;;FIND END OF MESSAGE
 9500  051660  001376                         BNE     2$
 9501  051662  163700  001224                 SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 9502  051666  006200                         ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 9503  051670  010037  001226                 MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 9504  051674  012737  000004  001210         MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 9505  051702  000413                         BR      5$
 9506  051704  017637  000004  051730  3$:    MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 9507  051712  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 9508  051720  013746  177776                 MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 9509  051724  004737  051046                 JSR     PC,$TYPE        ;;CALL TYPE MACRO
 9510  051730  000000                  4$:    .WORD   0
 9511  051732                           5$:
 9512  051732  105737  052020          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9513  051736  001416                         BEQ     12$             ;;IF NOT:  BR
 9514  051740  005737  001230                 TST     $ENV            ;;RUNNING UNDER APT?
 9515  051744  001413                         BEQ     12$             ;;IF NOT:  BR
 9516  051746  005737  001210          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9517  051752  001375                         BNE     11$             ;;IF NOT:  WAIT
 9518  051754  017637  000004  001212         MOV     @4(SP),$FATAL   ;;GET ERROR #
 9519  051762  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9520  051770  005237  001210                 INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9521  051774  105037  052020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 9522  052000  105037  052017                 CLRB    $LFLG           ;;CLEAR LOG FLAG
 9523  052004  105037  052016                 CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9524  052010  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 9525  052012  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 9526  052014  000207                         RTS     PC              ;;RETURN
 9527  052016     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
 9528  052017     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
 9529  052020     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
 9530          052022                         .EVEN
 9531          000200                 APTSIZE=200
 9532          000001                 APTENV=001
 9533          000100                 APTSPOOL=100
 9534          000040                 APTCSUP=040
 9535                                 .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 9536
 9537                                 ;;**************************************************************************
 9538                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 9539                                 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 9540                                 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 9541                                 ;*CALL:
 9542                                 ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9543                                 ;*      TYPOS                   ;;CALL FOR TYPEOUT
 9544                                 ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 9545                                 ;*      .BYTE   M               ;;M=1 OR 0
 9546                                 ;*                                      ;;1=TYPE LEADING ZEROS
 9547                                 ;*                                      ;;0=SUPPRESS LEADING ZEROS
 9548                                 ;*
 9549                                 ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
 9550                                   ;*$TYPOS OR $TYPOC
 9551                                   ;*CALL:
 9552                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9553                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
 9554                                   ;*
 9555                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 9556                                   ;*CALL:
 9557                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9558                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
 9559
 9560  052022  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 9561  052026  116637  000001  052245           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 9562  052034  112637  052247                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 9563  052040  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 9564  052044  000406                           BR      $TYPON
 9565  052046  112737  000001  052245  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 9566  052054  112737  000006  052247           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 9567  052062  112737  000005  052244  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 9568  052070  010346                           MOV     R3,-(SP)        ;;SAVE R3
 9569  052072  010446                           MOV     R4,-(SP)        ;;SAVE R4
 9570  052074  010546                           MOV     R5,-(SP)        ;;SAVE R5
 9571  052076  113704  052247                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 9572  052102  005404                           NEG     R4
 9573  052104  062704  000006                   ADD     #6.R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 9574  052110  110437  052246                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 9575  052114  113704  052245                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 9576  052120  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 9577  052124  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
 9578  052126  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 9579  052130  000404                           BR      3$              ;;GO DO MSB
 9580  052132  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
 9581  052134  006105                           ROL     R5
 9582  052136  006105                           ROL     R5
 9583  052140  010503                           MOV     R5,R3
 9584  052142  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 9585  052144  105337  052246                   DECB    $OMODE          ;;TYPE THIS DIGIT?
 9586  052150  100016                           BPL     7$              ;;BR IF NO
 9587  052152  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
 9588  052156  001002                           BNE     4$              ;;TEST FOR 0
 9589  052160  005704                           TST     R4              ;;SUPPRESS THIS 0?
 9590  052162  001403                           BEQ     5$              ;;BR IF YES
 9591  052164  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 9592  052166  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 9593  052172  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 9594  052176  110337  052242                   MOVB    R3,8$           ;;SAVE FOR TYPING
 9595  052202  104401  052242                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
 9596  052206  105337  052244           7$:     DECB    $OCNT           ;;COUNT BY 1
 9597  052212  003347                           BGT     2$              ;;BR IF MORE TO DO
 9598  052214  002402                           BLT     6$              ;;BR IF DONE
 9599  052216  005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 9600  052220  000744                           BR      2$              ;;GO DO THE LAST DIGIT
 9601  052222  012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
 9602  052224  012604                           MOV     (SP)+,R4        ;;RESTORE R4
 9603  052226  012603                           MOV     (SP)+,R3        ;;RESTORE R3
 9604  052230  016666  000002  000004           MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 9605  052236  012616                           MOV     (SP)+,(SP)
```

```
052240  000002                          RTI                             ;;RETURN
052242     000                  8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
052243     000                          .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
052244     000                  $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
052245     000                  $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
052246  000000                  $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*******************************************************************
                                .ENABL  LSB
052250  000000                  $TKCNT: .WORD   0                       ;;NUMBER OF ITEMS IN QUEUE
052252  000000                  $TKQIN: .WORD   0                       ;;INPUT POINTER
052254  000000                  $TKQOUT: .WORD  0                       ;;OUTPUT POINTER
052256  000001                  $TKQSRT: .BLKB  1                       ;;TTY KEYBOARD QUEUE
        052257                  $TKQEND=.
        052260                  .EVEN

                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN
                                ;
052260  005037  052250          $TKINT: CLR     $TKCNT                  ;;CLEAR COUNT OF ITEMS IN QUEUE
052264  012737  052256  052252          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
052272  013737  052252  052254          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
052300  012737  052330  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
052306  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
052314  005777  126626                  TST     @$TKB                   ;;CLEAR DONE FLAG
052320  012777  000100  126616          MOV     #100,@$TKS              ;;ENABLE TTY KEYBOARD INTERRUPT
052326  000207                          RTS     PC                      ;;RETURN TO CALLER

                                ;*TK SERVICE ROUTINE
                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                ;*IT IN THE QUEUE.
                                ;*IF THE CHARACTER IS A "CONTROL-C" ( C) $TKINT IS CALLED AND
                                ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
                                ;
052330  117746  126612          $TKSRV: MOVB    @$TKB,-(SP)             ;;PICKUP THE CHARACTER
052334  042716  177600                  BIC     # C177,(SP)             ;;STRIP THE JUNK
052340  021627  000003                  CMP     (SP),#3                 ;;IS IT A CONTROL C?
052344  001007                          BNE     1$                      ;;BRANCH IF NO
052346  104401  053456                  TYPE    ,$CNTLC                 ;;TYPE A CONTROL-C ( C)
052352  004737  052260                  JSR     PC,$TKINT               ;;INIT THE KEYBOARD
052356  005726                          TST     (SP)+                   ;;CLEAN UP STACK
052360  000137  047330                  JMP     STOP                    ;;CONTROL C RESTART
052364  021627  000007          1$:     CMP     (SP),#7                 ;;IS IT A CONTROL G?
052370  001004                          BNE     2$                      ;;BRANCH IF NO
052372  022737  000176  001140          CMP     #SWREG,SWR              ;;IS SOFT-SWR SELECTED?
052400  001500                          BEQ     6$                      ;;GO TO SWR CHANGE

052402                          2$.
052402  022737  000001  052250          CMP     #1,$TKCNT               ;;IS THE QUEUE FULL?
```

```
 9662  052410  001004                        BNE     3$              ;;BRANCH IF NO
 9663  052412  104401  001200                TYPE    ,$BELL          ;;RING THE TTY BELL
 9664  052416  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
 9665  052420  000451                        BR      5$              ;;EXIT
 9666  052422  021627  000023        3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
 9667  052426  001021                        BNE     32$             ;;BRANCH IF NO
 9668  052430  005077  126510                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 9669  052434  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
 9670  052436  105777  126502        31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
 9671  052442  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
 9672  052444  117746  126476                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
 9673  052450  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9674  052454  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 9675  052460  001366                        BNE     31$             ;;BRANCH IF NO
 9676  052462  012777  000100  126454        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
 9677  052470  000002                        RTI                     ;;RETURN
 9678  052472  005237  052250        32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
 9679  052476  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
 9680  052502  002405                        BLT     4$              ;;BRANCH IF YES
 9681  052504  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
 9682  052510  003002                        BGT     4$              ;;BRANCH IF YES
 9683  052512  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
 9684  052516  112677  177530        4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
 9685  052522  005237  052252                INC     $TKQIN          ;;UPDATE THE POINTER
 9686  052526  023727  052252  052257        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
 9687  052534  001003                        BNE     5$              ;;BRANCH IF NO
 9688  052536  012737  052256  052252        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
 9689  052544  000002                5$:     RTI                     ;;RETURN
 9690
 9691                                ;;**************************************************************
 9692                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 9693                                ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
 9694                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
 9695                                ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
 9696  052546  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
 9697  052554  001124                        BNE     15$             ;;EXIT IF NOT
 9698  052556  105777  126362                TSTB    @$TKS           ;;IS A CHAR WAITING?
 9699  052562  100121                        BPL     15$             ;;IF NOT, EXIT
 9700  052564  117746  126356                MOVB    @$TKB,-(SP)     ;;YES
 9701  052570  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
 9702  052574  021627  000007                CMP     (SP),#7         ;;IS IT A CONTROL-G?
 9703  052600  001300                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
 9704                                                                ;;AND EXIT
 9705
 9706                                ;;**************************************************************
 9707                                ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
 9708                                ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
 9709                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 9710  052602  123727  001134  000001 6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 9711  052610  001674                        BEQ     2$              ;;BRANCH IF YES
 9712  052612  005726                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
 9713  052614  004737  052260                JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
 9714  052620  005077  126320                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 9715  052624  112737  000001  001135        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
 9716
 9717  052632  104401  053470                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G ( G)
```

```
 9718  052636  104401  053475          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 9719  052642  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 9720  052646  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9721  052650  104401  053506                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 9722  052654  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 9723  052656  005046                          CLR     -(SP)           ;;THE NEW SWR
 9724  052660  105777  126260          7$:     TSTB    @$TKS           ;;CHAR THERE?
 9725  052664  100375                          BPL     7$              ;;IF NOT TRY AGAIN
 9726
 9727  052666  117746  126254                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 9728  052672  042716  177600                  BIC     # C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9729
 9730  052676  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
 9731  052702  001015                          BNE     9$              ;;BRANCH IF NOT
 9732  052704  104401  053456                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C ( C)
 9733  052710  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
 9734  052714  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
 9735  052722  001003                          BNE     8$              ;;BRANCH IF NO
 9736  052724  012777  000100  126212          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
 9737  052732  000137  047330          8$:     JMP     STOP            ;;CONTROL-C RESTART
 9738
 9739
 9740  052736  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 9741  052742  001005                          BNE     10$             ;;BRANCH IF NOT
 9742  052744  104401  053463                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U ( U)
 9743  052750  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 9744  052754  000737                          BR      19$             ;;LET'S TRY IT AGAIN
 9745
 9746
 9747  052756  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
 9748  052762  001022                          BNE     16$             ;;BRANCH IF NO
 9749  052764  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 9750  052770  001403                          BEQ     11$             ;;BRANCH IF YES
 9751  052772  016677  000002  126140          MOV     2(SP),@SWR      ;;SAVE NEW SWR
 9752  053000  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
 9753  053004  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 9754  053010  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 9755  053016  001003                          BNE     15$             ;;BRANCH IF NOT
 9756  053020  012777  000100  126116          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD :NTERRUPTS
 9757  053026  000002                  15$:    RTI                     ;;RETURN
 9758  053030  004737  051260          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 9759  053034  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
 9760  053040  002420                          BLT     18$             ;;BRANCH IF YES
 9761  053042  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
 9762  053046  003015                          BGT     18$             ;;BRANCH IF YES
 9763  053050  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 9764  053054  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
 9765  053060  001403                          BEQ     17$             ;;BRANCH IF YES
 9766  053062  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
 9767  053064  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
 9768  053066  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
 9769  053070  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 9770  053074  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 9771  053100  000667                          BR      7$              ;;GET THE NEXT ONE
 9772  053102  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 9773  053106  000720                          BR      20$             ;;SIMULATE CONTROL-U
```

```
                                .DSABL  LSB


                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
053110  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
053112  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
053120  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
053124  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
053126  012746  053134                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
053132  000002                          RTI                     ;;POP NEW PC AND PS
053134                          64$:
053134  005737  052250          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
053140  001775                          BEQ     1$
053142  005337  052250                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
053146  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
053154  005237  052254                  INC     $TKQOUT         ;;UPDATE THE POINTER
053160  023727  052254  052257          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
053166  001003                          BNE     2$              ;;BRANCH IF NO
053170  012737  052256  052254          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
053176  000002                  2$:     RTI                     ;;RETURN
                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

053200  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
053202  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
053204  012703  053434          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
053210  022703  053456          2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
053214  101456                          BLOS    4$              ;;BR IF YES
053216  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
053220  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
053222  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
053226  001022                          BNE     5$              ;;BR IF NO
053230  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
053232  001007                          BNE     6$              ;;BR IF NO
053234  112737  000134  053432          MOVB    #'\,9$          ;;TYPE A BACK SLASH
053242  104401  053432                  TYPE    ,3$
053246  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
053252  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
053254  020327  053434                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
053260  103434                          BLO     4$              ;;BR IF YES
053262  111337  053432                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
053266  104401  053432                  TYPE    ,9$             ;;GO TYPE
053272  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
053274  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
053276  001406                          BEQ     7$              ;;BR IF NO
```

```
 9830  053300  112737  000134  053432          MOVB    #' ,9$          ;;TYPE A BACK SLASH
 9831  053306  104401  053432                  TYPE    ,9$
 9832  053312  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
 9833  053314  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
 9834  053320  001003                          BNE     8$              ;;BR IF NO
 9835  053322  104401  053463                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
 9836  053326  000726                          BR      1$              ;;GO START OVER
 9837  053330  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A " R"?
 9838  053334  001011                          BNE     3$              ;;BRANCH IF NO
 9839  053336  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
 9840  053340  104401  001205                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
 9841  053344  104401  053434                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
 9842  053350  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
 9843  053352  104401  001204          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 9844  053356  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 9845  053360  111337  053432          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 9846  053364  104401  053432                  TYPE    ,9$
 9847  053370  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 9848  053374  001305                          BNE     2$              ;;LOOP IF NOT RETURN
 9849  053376  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 9850  053402  104401  001206                  TYPE    ,$LF            ;;TYPE A LINE FEED
 9851  053406  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 9852  053410  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 9853  053412  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 9854  053414  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 9855  053422  012766  053434  000004          MOV     #$TTYIN,4(SP)
 9856  053430  000002                          RTI                     ;;RETURN
 9857  053432     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR.  TO TYPE
 9858  053433     000                  .       .BYTE   0               ;;TERMINATOR
 9859  053434  000022          $TTYIN: .BLKB   22                      ;;RESERVE 22 BYTES FOR TTY INPUT
 9860  053456  041536  005015     000  $CNTLC: .ASCIZ  / C/<15><12>    ;;CONTROL "C"
 9861  053463     136  006525  000012  $CNTLU: .ASCIZ  / U/<15><12>    ;;CONTROL "U"
 9862  053470  043536  005015     000  $CNTLG: .ASCIZ  / G/<15><12>    ;;CONTROL "G"
 9863  053475     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
 9864  053502  036440  000040
 9865  053506  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
 9866  053514  020075     000
 9867          053520                  .EVEN
 9868                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 9869
 9870                                  ;;**************************************************************
 9871                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 9872                                  ;*CHANGE IT TO BINARY.
 9873                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
 9874                                  ;*OCTAL DIGITS.  IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
 9875                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED.  THE COMPLETE NUMBER MUST
 9876                                  ;*THEN BE RETYPED.  THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
 9877                                  ;*CALL:
 9878                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 9879                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 9880                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 9881
 9882  053520  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 9883  053522  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
 9884  053530  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9885  053532  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
 9886  053534  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9887  053536  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
 9888  053540  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 9889  053542  010037  053646          MOV     R0,5$           ;;AND SAVE IT
 9890  053546  005001                  CLR     R1              ;;CLEAR DATA WORD
 9891  053550  005002                  CLR     R2
 9892  053552  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 9893  053554  001420                  BEQ     3$              ;;IF ZERO GET OUT
 9894  053556  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
 9895  053562  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
 9896  053564  122716  000067          CMPB    #'7,(SP)
 9897  053570  002423                  BLT     4$
 9898  053572  006301                  ASL     R1              ;;*2
 9899  053574  006102                  ROL     R2
 9900  053576  006301                  ASL     R1              ;;*4
 9901  053600  006102                  ROL     R2
 9902  053602  006301                  ASL     R1              ;;*8
 9903  053604  006102                  ROL     R2
 9904  053606  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
 9905  053612  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
 9906  053614  000756                  BR      2$              ;;LOOP
 9907  053616  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
 9908  053620  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
 9909  053624  010237  053656          MOV     R2,$HIOCT
 9910  053630  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 9911  053632  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 9912  053634  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
 9913  053636  000002                  RTI                     ;;RETURN
 9914  053640  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
 9915  053642  105010                  CLRB    (R0)            ;;SET A TERMINATOR
 9916  053644  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
 9917  053646  000000          5$:     .WORD   0
 9918  053650  104401  001204          TYPE    ,$QUES          ;;"?" "CR" & "LF"
 9919  053654  000730                  BR      1$              ;;TRY AGAIN
 9920  053656  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
 9921                          .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
 9922
 9923                          ;;*****************************************************************
 9924                          ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
 9925                          ;*UNSIGNED OCTAL ASCIZ NUMBER.
 9926                          ;*CALL
 9927                          ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 9928                          ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
 9929                          ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
 9930
 9931
 9932  053660  104413          $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
 9933  053662  016601  000002          MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
 9934  053666  012705  053777          MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
 9935  053672  012704  000014          MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
 9936  053676  012703  177770          MOV     #^C7,R3         ;;MASK
 9937  053702  012100                  MOV     (R1)+,R0        ;;LOWER WORD
 9938  053704  012101                  MOV     (R1)+,R1        ;;HIGH WORD
 9939  053706  005002                  CLR     R2              ;;TERMINATOR
 9940  053710  110245          1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
 9941  053712  010002                  MOV     R0,R2           ;;GET THIS DIGIT
```

```
 9942  053714  005304                          DEC     R4              ;;COUNT THIS CHARACTER
 9943  053716  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
 9944  053720  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
 9945  053722  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
 9946  053724  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
 9947  053730  104414                          RESREG                  ;;RESTORE ALL REGISTERS
 9948  053732  000207                          RTS     PC              ;;RETURN TO USER
 9949  053734  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
 9950  053736  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
 9951  053740  006000                          ROR     R0      !       ;;      THE NEXT OCTAL DIGIT
 9952  053742  006001                          ROR     R1
 9953  053744  006000                          ROR     R0
 9954  053746  006001                          ROR     R1
 9955  053750  006000                          ROR     R0
 9956  053752  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
 9957  053754  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
 9958  053760  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
 9959  053762  000016                  $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
 9960                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
 9961
 9962                                  ;;**************************************************************
 9963                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
 9964                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
 9965                                  ;*POSITIVE.
 9966                                  ;*CALL
 9967                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 9968                                  ;*      JSR     PC,@#$DB2D
 9969                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
 9970                                                                  ;;IS ON THE STACK
 9971
 9972
 9973  054000  104413                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
 9974  054002  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
 9975  054006  012700  054160                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
 9976  054012  010066  000002                  MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
 9977  054016  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
 9978  054020  012202                          MOV     (R2)+,R2
 9979  054022  012737  000012  054076          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
 9980  054030  012704  054110                  MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
 9981  054034  012705  054112                  MOV     #$TNPWR+2,R5
 9982  054040  005003                  1$:     CLR     R3              ;;CLEAR PARTIAL
 9983  054042  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
 9984  054044  005602                          SBC     R2
 9985  054046  161502                          SUB     (R5),R2
 9986  054050  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
 9987  054052  005203                          INC     R3              ;;ADD 1 TO PARTIAL
 9988  054054  000772                          BR      2$              ;;LOOP
 9989  054056  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
 9990  054060  005502                          ADC     R2
 9991  054062  062402                          ADD     (R4)+,R2
 9992  054064  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
 9993  054066  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
 9994  054072  110320                          MOVB    R3,(R0)+        ;;SAVE IT
 9995  054074  005327                          DEC     (PC)+           ;;DONE?
 9996  054076  000000                  4$:     .WORD   0
 9997  054100  001357                          BNE     1$              ;;BR IF NO
```

```
 9998  054102  105020                        CLRB    (R0)+           ;;TERMINATOR
 9999  054104  104414                        RESREG                  ;;RESTORE REGISTERS
10000  054106  000207                        RTS     PC              ;;RETURN
10001  054110  145000                $TNPWR. 145000                  ;;1.0E09
10002  054112  035632                        35632
10003  054114  160400                        160400                  ;;1.0E08
10004  054116  002765                        2765
10005  054120  113200                        113200                  ;;1.0E07
10006  054122  000230                        230
10007  054124  041100                        041100                  ;;1.0E06
10008  054126  000017                        17
10009  054130  103240                        103240                  ;;1.0E05
10010  054132  000001                        1
10011  054134  023420                        23420                   ;;1.0E04
10012  054136  000000                        0
10013  054140  001750                        1750                    ;;1.0E03
10014  054142  000000                        0
10015  054144  000144                        144                     ;;1.0E02
10016  054146  000000                        0
10017  054150  000012                        12                      ;;1.0E01
10018  054152  000000                        0
10019  054154  000001                        1                       ;;1.0E00
10020  054156  000000                        0
10021  054160  000014                $DECVL: .BLKB   12              ;;RESERVE STORAGE FOR ASCIZ STRING
10022                                  .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
10023
10024                                  ;;**************************************************************
10025                                  ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
10026                                  ;*UNSIGNED DECIMAL ASCIZ NUMBER.
10027                                  ;*CALL
10028                                  ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
10029                                  ;*      JSR     PC,@#$SB2D      ;;CALL
10030                                  ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR. IS ON THE STACK
10031
10032
10033  054174  016637  000002  054224 $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
10034  054202  012746  054224                MOV     #1$,-(SP)       ;;SET POINTER
10035  054206  004737  054000                JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
10036  054212  062716  000005                ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
10037  054216  012666  000002                MOV     (SP)+,2(SP)     ;;PICKUP POINTER
10038  054222  000207                        RTS     PC              ;;RETURN
10039  054224  000000  000000          1$:   .WORD   0,0
10040                                  .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
10041
10042                                  ;;**************************************************************
10043                                  ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
10044                                  ;*LEADING NUMBERS.
10045                                  ;*CALL
10046                                  ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
10047                                  ;*      JSR     PC,@#$SUPRS
10048
10049
10050  054230  010046                $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
10051  054232  016600  000004                MOV     4(SP),R0        ;;PICKUP THE POINTER
10052  054236  105710                  1$:   TSTB    (R0)            ;;TERMINATEOR?
10053  054240  001403                        BEQ     2$              ;;BR IF YES
```

```
10054  054242  122720  000060                  CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
10055  054246  001773                          BEQ     1$              ;;BR IF YES
10056  054250  005300                  2$:     DEC     R0              ;;BACKUP BY "1"
10057  054252  010037  054260                  MOV     R0,3$           ;;SAVE FOR TYPING
10058  054256  104401                          TYPE                    ;;GO TYPE
10059  054260  000000                  3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
10060  054262  012600                          MOV     (SP)+,R0        ;;RESTORE R0
10061  054264  012616                          MOV     (SP)+,(SP)      ;;RESTORE THE STACK
10062  054266  000207                          RTS     PC              ;;RETURN
10063                                  .SBTTL  INTEGER MULTIPLY ROUTINE
10064
10065                                  ;;**************************************************************************
10066                                  ;*CALL
10067                                  ;*      MOV     MULTIPLER,-(SP)
10068                                  ;*      MOV     MULTIPLICAND,-(SP)
10069                                  ;*      JSR     PC,@#$MULT
10070                                  ;*      RETURN  ;;PRODUCT IS ON THE STACK
10071                                  ;*
10072                                  ;*      STACK   PRODUCT
10073                                  ;*      -----   -------
10074                                  ;*      TOP     LSB'S
10075                                  ;*      +2      MSB'S
10076
10077  054270                          $MULT:
10078  054270  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10079  054272  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10080  054274  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10081  054276  005046                          CLR     -(SP)           ;;CLEAR THE SIGN KEY
10082  054300  016601  000012                  MOV     12(SP),R1       ;;GET THE MULTIPLICAND
10083  054304  100002                          BPL     1$              ;;BR IF PLUS
10084  054306  005216                          INC     (SP)            ;;SET THE SIGN KEY
10085  054310  005401                          NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
10086  054312  016602  000014          1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
10087  054316  100002                          BPL     2$              ;;BR IF PLUS
10088  054320  005316                          DEC     (SP)            ;;UPDATE THE SIGN KEY
10089  054322  005402                          NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
10090  054324  012746  000021          2$:     MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
10091  054330  005000                          CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
10092  054332  103001          3$:     BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
10093  054334  060200                          ADD     R2,R0
10094  054336  006000          4$:     ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
10095  054340  006001                          ROR     R1              ;;THE MULTIPLICAND
10096  054342  005316                          DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
10097  054344  001372                          BNE     3$              ;;BR IF NO
10098  054346  022616                          CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
10099  054350  001403                          BEQ     5$              ;;GO TO EXIT IF NO
10100  054352  005400                          NEG     R0              ;;YES--SO MAKE IT SO
10101  054354  005401                          NEG     R1
10102  054356  005600                          SBC     R0
10103  054360  005726          5$:     TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
10104  054362  010066  000012                  MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
10105  054366  010166  000010                  MOV     R1,10(SP)       ;;LSB'S
10106  054372  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10107  054374  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10108  054376  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10109  054400  000207                          RTS     PC
```

```
                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                ;;**********************************************************************
                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*      SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+14---R0

054402                          $SAVREG:
054402  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
054404  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
054406  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
054410  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
054412  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
054414  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
054416  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
054422  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
054426  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
054432  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
054436  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
054440                          $RESREG:
054440  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
054444  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
054450  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
054454  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
054460  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
054462  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
054464  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
054466  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
054470  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
054472  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
054474  000002                          RTI
                                .SBTTL  TRAP DECODER

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

054476  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
054500  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
054504  005740                          TST     -(R0)           ;;BACKUP BY 2
```

```
10166  054506  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
10167  054510  006300                          ASL     R0              ;;POSITION FOR INDEXING
10168  054512  016000  054532                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
10169  054516  000200                          RTS     R0              ;;GO TO ROUTINE
10170
10171
10172                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
10173
10174  054520  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
10175  054522  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
10176  054530  000002                          RTI                     ;;RESTORE THE PSW
10177
10178                                  .SBTTL  TRAP TABLE
10179
10180                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10181                                  ;*BY THE "TRAP" INSTRUCTION.
10182
10183                                  ;       ROUTINE
10184                                  ;       -------
10185  054532  054520                  $TRPAD: .WORD   $TRAP2
10186  054534  051046                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
10187  054536  052046                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10188  054540  052022                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
10189  054542  052062                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
10190  054544  051330                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
10191
10192  054546  052636                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
10193
10194  054550  052546                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
10195  054552  053110                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
10196  054554  053200                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
10197  054556  053520                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
10198  054560  054402                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
10199  054562  054440                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
10200  054564  047276                          SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
10201
```

```
10202
10203                                   .SBTTL  SERVICE MESSAGES
10204
10205  054566  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC/
10206  054574  051525  051040  030113
10207  054602  026466  045522  033460
10208  054610  042040  040525  020114
10209  054616  047520  052122  042040
10210  054624  044522  042526  042040
10211  054632  040511  047107  051517
10212  054640  044524     103
10213  054643     015  046412  044501          .ASCII  <CR><LF>/MAINDEC-11-DZR6G-A-PB/<CR><LF>
10214  054650  042116  041505  030455
10215  054656  026461  055104  033122
10216  054664  026507  026501  041120
10217  054672  005015
10218  054674  005015  025011  025052          .ASCII  <CR><LF>/       ***** CAUTION *****/<CR><LF>
10219  054702  025052  041440  052501
10220  054710  044524  047117  025040
10221  054716  025052  025052  005015
10222  054724  005015  044124  051511          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY BY TYPING CONTROL-C/
10223  054732  050040  047522  051107
10224  054740  046501  051440  047510
10225  054746  046125  020104  041040
10226  054754  020105  040510  052114
10227  054762  042105  047440  046116
10228  054770  020131  054502  052040
10229  054776  050131  047111  020107
10230  055004  047503  052116  047522
10231  055012  026514     103
10232  055015     015  047412  044124              .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND, OR THE DRIVE/
10233  055022  051105  044527  042523
10234  055030  020054  040503  052122
10235  055036  044522  043504  020105
10236  055044  047506  046522  052101
10237  055052  044524  043516  040440
10238  055060  042116  020054  051117
10239  055066  052040  042510  042040
10240  055074  044522  042526
10241  055100  005015  040515  020131          .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/
10242  055106  042502  046040  043105
10243  055114  020124  047111  040440
10244  055122  020116  047125  042504
10245  055130  042524  046522  047111
10246  055136  042105  051440  040524
10247  055144  042524
10248  055146  005015  047111  052111          .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
10249  055154  040511  046114  026131
10250  055162  042040  044522  042526
10251  055170  020123  047524  041040
10252  055176  020105  042524  052123
10253  055204  042105  051440  047510
10254  055212  046125  020104  040510
10255  055220  042526  006472     012
10256  055225     015  040412  020056          .ASCII  <CR><LF>/A. EVEN NUMBERED UNIT SELECT PLUGS ONLY/
10257  055232  053105  047105  047040
```

```
10258  055240  046525  042502  042522
10259  055246  020104  047125  052111
10260  055254  051440  046105  041505
10261  055262  020124  046120  043525
10262  055270  020123  047117  054514
10263  055276  005015  027102  044040          .ASCII  <CR><LF>/B. HEADS MANUALLY LOADED/
10264  055304  040505  051504  046440
10265  055312  047101  040525  046114
10266  055320  020131  047514  042101
10267  055326  042105
10268  055330  005015  027103  041040          .ASCII  <CR><LF>/C. BOTH PORTS SELECTED/
10269  055336  052117  020110  047520
10270  055344  052122  020123  042523
10271  055352  042514  052103  042105
10272  055360  005015  027104  042040          .ASCII  <CR><LF>/D. DUAL PORT TEST SWITCH ENABLED/
10273  055366  040525  020114  047520
10274  055374  052122  052040  051505
10275  055402  020124  053523  052111
10276  055410  044103  042440  040516
10277  055416  046102  042105
10278  055422  005015  027105  053440          .ASCII  <CR><LF>/E. WRITE LOCK DISABLED/
10279  055430  044522  042524  046040
10280  055436  041517  020113  044504
10281  055444  040523  046102  042105
10282  055452  005015  027106  042040          .ASCII  <CR><LF>/F. DRIVE READY INDICATOR ON/<CR><LF>
10283  055460  044522  042526  051040
10284  055466  040505  054504  044440
10285  055474  042116  041511  052101
10286  055502  051117  047440  006516
10287  055510     012
10288  055511     015  042012  044522          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
10289  055516  042526  020123  047516
10290  055524  020124  047524  041040
10291  055532  020105  042524  052123
10292  055540  042105  046440  051525
10293  055546  020124  040510  042526
10294  055554  005015  047502  044124          .ASCII  <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
10295  055562  050040  051117  051524
10296  055570  042040  051505  046105
10297  055576  041505  042524  000504
10298  055604     012
10299  055605     015  042012  051511          .ASCII  <CR><LF>/DISABLE DUAL PORT TEST SWITCH ON DUAL PORT MODULE/
10300  055612  041101  042514  042040
10301  055620  040525  020114  047520
10302  055626  052122  052040  051505
10303  055634  020124  053523  052111
10304  055642  044103  047440  020116
10305  055650  052504  046101  050040
10306  055656  051117  020124  047515
10307  055664  052504  042514
10308  055670  005015  042502  047506          .ASCIZ  <CR><LF>/BEFORE PROCEEDING TO ANY OTHER PROGRAM/<CR><LF>
10309  055676  042522  050040  047122
10310  055704  042503  042105  047111
10311  055712  020107  047524  040440
10312  055720  054516  047440  044124
10313  055726  051105  050040  047522
```

```
10314   055734  051107  046501  005015
10315   055742     000
10316   055743     015  041012  020105  MSG2:   .ASCIZ  <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
10317   055750  052523  042522  052040
10318   055756  020117  052520  020124
10319   055764  041523  040522  041524
10320   055772  020110  040520  045503
10321   056000  044440  020116  051104
10322   056006  053111  020105  000060
10323   056014  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED (EVEN NOS. ONLY): /
10324   056022  024105  024523  052040
10325   056030  020117  042502  052040
10326   056036  051505  042524  020104
10327   056044  042450  042526  020116
10328   056052  047516  027123  047440
10329   056060  046116  024531  020072
10330   056066     000
10331   056067     015  041012  051525  MSG4:   .ASCIZ  <CR><LF>/BUSS ADDRESS (177440): /
10332   056074  020123  042101  051104
10333   056102  051505  020123  030450
10334   056110  033467  032064  024460
10335   056116  020072     000
10336   056121     015  041412  047117  MSG5:   .ASCIZ  <CR><LF>/CONTROLLER INTERRUPT VECTOR (210): /
10337   056126  051124  046117  042514
10338   056134  020122  047111  042524
10339   056142  051122  050125  020124
10340   056150  042526  052103  051117
10341   056156  024040  030462  024460
10342   056164  020072     000
10343   056167     015  044412  052116  MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
10344   056174  051105  052522  052120
10345   056202  047440  041503  051125
10346   056210  042522  020104  052101
10347   056216  050040  036503     000
10348   056223     015  042012  044522  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
10349   056230  042526  030040  053440
10350   056236  046111  020114  047516
10351   056244  020124  042502  052040
10352   056252  051505  042524  000104
10353   056260  005015  005015  046101  MSG8:   .ASCIZ  <CR><LF><CR><LF>/ALL DRIVES TESTED/<CR><LF><LF>
10354   056266  020114  051104  053111
10355   056274  051505  052040  051505
10356   056302  042524  006504  005012
10357   056310     000
10358   056311     015  005012  044527  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES: /
10359   056316  046114  052040  051505
10360   056324  020124  051104  053111
10361   056332  051505  000072
10362   056336  005015  050012  053517  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
10363   056344  051105  052440  020120
10364   056352  042522  052123  051101
10365   056360  020124  047524  052040
10366   056366  051505  020124  006461
10367   056374  000012
10368   056376  005015  047516  046040  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
10369   056404  047440  020122  020120
```

```
10370  056412  046103  041517  051513
10371  056420  050040  042522  042523
10372  056426  052116
10373  056430  005015  046101  020114          .ASCIZ  <CR><LF>/ALL TESTS BYPASSED/
10374  056436  042524  052123  020123
10375  056444  054502  040520  051523
10376  056452  042105     000
10377  056455     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
10378  056462  051501  044523  043516
10379  056470  042040  044522  042526
10380  056476  000040
10381  056500  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
10382  056506  042526  000040
10383  056512  005015  042012  044522  MSG16:  .ASCIZ  <CR><LF><LF>/DRIVE SERIAL NO. /
10384  056520  042526  051440  051105
10385  056526  040511  020114  047516
10386  056534  020056     000
10387  056537     015  041412  051101  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
10388  056544  051124  042111  042507
10389  056552  051440  051105  040511
10390  056560  020114  047516  020056
10391  056566     000
10392  056567     015  050012  051117  MSG19:  .ASCII  <CR><LF>/PORT /
10393  056574  020124
10394  056576  006440  000012          MSG19A: .ASCIZ  / /<CR><LF>
10395  056602  005015  047520  052122  MSG20:  .ASCIZ  <CR><LF>/PORT A TIMEOUT: /
10396  056610  040440  052040  046511
10397  056616  047505  052125  020072
10398  056624     000
10399  056625     015  040412  047502  MSG21:  .ASCIZ  <CR><LF>/ABORTING BALANCE OF TESTS/<CR><LF>
10400  056632  052122  047111  020107
10401  056640  040502  040514  041516
10402  056646  020105  043117  052040
10403  056654  051505  051524  005015
10404  056662     000
10405  056663     040  051515     000  MSG22:  .ASCIZ  / MS/
10406  056667     015  050012  051117  MSG23:  .ASCIZ  <CR><LF>/PORT B TIMEOUT: /
10407  056674  020124  020102  044524
10408  056702  042515  052517  035124
10409  056710  000040
10410  056712  005015  051120  043517  MSG74:  .ASCIZ  <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
10411  056720  040522  020115  041101
10412  056726  051117  020124  042520
10413  056734  042116  047111  027107
10414  056742  027056  046120  040505
10415  056750  042523  053440  044501
10416  056756  000124
10417  056760  005015  040510  052114  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING...PLEASE WAIT/
10418  056766  050040  047105  044504
10419  056774  043516  027056  050056
10420  057002  042514  051501  020105
10421  057010  040527  052111     000
10422  057015     015  050012  047522  MSG76:  .ASCIZ  <CR><LF>/PROGRAM ABORTED/
10423  057022  051107  046501  040440
10424  057030  047502  052122  042105
10425  057036     000
```

```
10426  057037  015     041412  052520  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
10427  057044  044040  046101  042524
10428  057052  000104
10429
10430                                  .SBTTL  ERROR MESSAGES
10431
10432  057054  005015  047523  051122  EM1:    .ASCIZ  <CR><LF>/SORRY, ONLY 0,2,4,6 ALLOWED, TRY AGAIN/<CR><LF>
10433  057062  026131  047440  046116
10434  057070  020131  026060  026062
10435  057076  026064  020066  046101
10436  057104  047514  042527  026104
10437  057112  052040  054522  040440
10438  057120  040507  047111  005015
10439  057126  000
10440  057127  104     044522  042526  EM2:    .ASCIZ  /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
10441  057134  021440  044440  020116
10442  057142  045522  051503  020062
10443  057150  040503  047116  052117
10444  057156  041040  020105  042522
10445  057164  042101  041040  041501
10446  057172  020113  047503  051122
10447  057200  041505  046124  020131
10448  057206  047111  051040  046513
10449  057214  031122  000
10450  057217  015     040412  047502  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
10451  057224  052122  052040  051505
10452  057232  051524  027056  052456
10453  057240  042516  050130  041505
10454  057246  042524  020104  044524
10455  057254  042515  047440  052125
10456  057262  040440  020124  041520
10457  057270  000075
10458  057272  005015  041101  051117  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
10459  057300  020124  042524  052123
10460  057306  027123  027056  047125
10461  057314  054105  042520  052103
10462  057322  042105  044440  052116
10463  057330  051105  052522  052120
10464  057336  040440  020124  041520
10465  057344  000075
10466  057346  042115  020123  042523  EM5:    .ASCIZ  /MDS SET IN RKCS2/
10467  057354  020124  047111  051040
10468  057362  041513  031123  000
10469  057367  125     042506  051440  EM6:    .ASCIZ  /UFE SET IN RKCS2/
10470  057374  052105  044440  020116
10471  057402  045522  051503  000062
10472  057410  047516  042040  040522  EM7:    .ASCII  /NO DRA IN RKDS & NO NED IN RKCS2/
10473  057416  044440  020116  045522
10474  057424  051504  023040  047040
10475  057432  020117  042516  020104
10476  057440  047111  051040  041513
10477  057446  031123
10478  057450  005015  044103  041505          .ASCII  <CR><LF>/CHECK 1.  THE PORT A USING DRIVE FLOP (DP1)/
10479  057456  020113  027061  052040
10480  057464  042510  050040  051117
10481  057472  020124  020101  051525
```

```
10482  057500  047111  020107  051104
10483  057506  053111  020105  046106
10484  057514  050117  024040  050104
10485  057522  024461
10486  057524  005015  027062  042040          .ASCIZ  <CR><LF>/2. DRIVE AVAILABLE TO PORT A STATUS (DP2)/
10487  057532  044522  042526  040440
10488  057540  040526  046111  041101
10489  057546  042514  052040  020117
10490  057554  047520  052122  040440
10491  057562  051440  040524  052524
10492  057570  020123  042050  031120
10493  057576  000051
10494  057600  051104  053111  020105  EM8:     .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
10495  057606  051120  051505  047105
10496  057614  020124  052502  020124
10497  057622  047516  020124  050123
10498  057630  041505  043111  042511
10499  057636  020104  054502  047440
10500  057644  042520  040522  047524
10501  057652  000122
10502  057654  051104  053111  020105  EM9:     .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
10503  057662  047516  020124  051120
10504  057670  051505  047105  020124
10505  057676  052502  020124  050123
10506  057704  041505  043111  042511
10507  057712  020104  054502  047440
10508  057720  042520  040522  047524
10509  057726  000122
10510  057730  041101  051117  020124  EM10:    .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
10511  057736  042524  052123  027123
10512  057744  027056  040503  047116
10513  057752  052117  051040  043105
10514  057760  051105  047105  042503
10515  057766  041440  047117  051124
10516  057774  046117  042514  020122
10517  060002  042522  044507  052123
10518  060010  051105     000
10519  060013     104  040522  044440  EM11:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
10520  060020  020116  040522  051504
10521  060026  023040  047040  042105
10522  060034  044440  020116  045522
10523  060042  051503  020062  047502
10524  060050  044124  051440  052105
10525  060056     000
10526  060057     103  047117  051124  EM12:    .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
10527  060064  046117  042514  020122
10528  060072  047516  020124  042522
10529  060100  042101  020131  047111
10530  060106  051040  041513  030523
10531  060114     000
10532  060115     116  020117  052101  EM13:    .ASCIZ  /NO ATTN IN RKASOF/
10533  060122  047124  044440  020116
10534  060130  045522  051501  043117
10535  060136     000
10536  060137     125  042516  050130  EM14:    .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
10537  060144  041505  042524  020104
```

```
10538  060152  042515  047515  054522
10539  060160  050040  051101  052111
10540  060166  020131  051124  050101
10541  060174     000
10542  060175     122  042113  020103  EM15:   .ASCII  /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
10543  060202  020046  045522  040504
10544  060210  044440  042116  041511
10545  060216  052101  020105  044124
10546  060224  052101  053440  042503
10547  060232  047440  041503  051125
10548  060240  042522  020104  052101
10549  060246  005015  054503  020114          .ASCIZ  <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
10550  060254  030464  026061  052040
10551  060262  040522  045503  031040
10552  060270  020054  042523  052103
10553  060276  051117  031040  000061
10554  060304  040503  047116  052117  EM16:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
10555  060312  051040  040505  020104
10556  060320  040502  020104  042523
10557  060326  052103  051117  044440
10558  060334  043116  051117  040515
10559  060342  044524  047117     000
10560  060347     115  051505  040523  EM17:   .ASCIZ  /MESSAGE A0 ERROR/
10561  060354  042507  040440  020060
10562  060362  051105  047522  000122
10563  060370  042515  051523  043501  EM18:   .ASCIZ  /MESSAGE B0 ERROR/
10564  060376  020105  030102  042440
10565  060404  051122  051117     000
10566  060411     115  051505  040523  EM19:   .ASCIZ  /MESSAGE A1 ERROR/
10567  060416  042507  040440  020061
10568  060424  051105  047522  000122
10569  060432  042515  051523  043501  EM20:   .ASCIZ  /MESSAGE B1 ERROR/
10570  060440  020105  030502  042440
10571  060446  051122  051117     000
10572  060453     103  051105  020122  EM21:   .ASCIZ  /CERR SET IN RKCS1/
10573  060460  042523  020124  047111
10574  060466  051040  041513  030523
10575  060474     000
10576  060475     116  020117  051104  EM22:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/<CR><LF>
10577  060502  053111  051505  043040
10578  060510  052517  042116  044440
10579  060516  020116  042504  044526
10580  060524  042503  046440  050101
10581  060532  024040  042044  053105
10582  060540  024515  005015
10583  060544  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY AND RESTART/<CR><LF>
10584  060552  047503  051122  041505
10585  060560  046124  020131  047101
10586  060566  020104  042522  052123
10587  060574  051101  006524  000012
10588  060602  047516  042040  044522  EM23:   .ASCII  /NO DRIVES FOUND ON BUSS/<CR><LF>
10589  060610  042526  020123  047506
10590  060616  047125  020104  047117
10591  060624  041040  051525  006523
10592  060632     012
10593  060633     123  052105  050125          .ASCIZ  /SETUP CORRECTLY AND PRESS 'CONTINUE'/<CR><LF>
```

```
10594   060640  041440  051117  042522
10595   060646  052103  054514  040440
10596   060654  042116  050040  042522
10597   060662  051523  023440  047503
10598   060670  052116  047111  042525
10599   060676  006447  000012
10600   060702  047526  020114  040526  EM24:   .ASCIZ  /VOL VALID NOT SET IN RKMR2/
10601   060710  044514  020104  047516
10602   060716  020124  042523  020124
10603   060724  047111  051040  046513
10604   060732  031122     000
10605   060735     015  042012  052105  EM25:   .ASCIZ  <CR><LF>/DETECTED 10 BAD SECTORS...ABORTING TEST./
10606   060742  041505  042524  020104
10607   060750  030061  041040  042101
10608   060756  051440  041505  047524
10609   060764  051522  027056  040456
10610   060772  047502  052122  047111
10611   061000  020107  042524  052123
10612   061006     000
10613   061007     104  052105  041505  EM26:   .ASCIZ  /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
10614   061014  042524  020104  051502
10615   061022  020105  052502  020124
10616   061030  047516  020124  044514
10617   061036  052123  042105  044440
10618   061044  020116  040502  020104
10619   061052  042523  052103  051117
10620   061060  043040  046111  000105
10621   061066  042504  042524  052103  EM27:   .ASCII  /DETECTED BSE IN READ COMMAND/
10622   061074  042105  041040  042523
10623   061102  044440  020116  042522
10624   061110  042101  041440  046517
10625   061116  040515  042116
10626   061122  005015  052502  020124          .ASCIZ  <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
10627   061130  047516  020124  047111
10628   061136  050040  042522  044526
10629   061144  052517  020123  051127
10630   061152  052111  020105  047503
10631   061160  046515  047101  020104
10632   061166  047524  051440  046501
10633   061174  020105  042523  052103
10634   061202  051117     000
10635   061205     117  046116  020131  EM28:   .ASCII  /ONLY DRIVES 0,2,4,6 ALLOWED IN $DEVM/
10636   061212  051104  053111  051505
10637   061220  030040  031054  032054
10638   061226  033054  040440  046114
10639   061234  053517  042105  044440
10640   061242  020116  042044  053105
10641   061250     115
10642   061251     015  051012  046105          .ASCIZ  <CR><LF>/RELOAD $DEVM & PRESS CONTINUE./<CR><LF>
10643   061256  040517  020104  042044
10644   061264  053105  020115  020046
10645   061272  051120  051505  020123
10646   061300  047503  052116  047111
10647   061306  042525  005015     000
10648   061313     120  051117  020124  EM29:   .ASCIZ  /PORT AVAILABLE...TIMERS NOT INHIBITED/
10649   061320  053101  044501  040514
```

```
10650  061326  046102  027105  027056
10651  061334  044524  042515  051522
10652  061342  047040  052117  044440
10653  061350  044116  041111  052111
10654  061356  042105     000
10655  061361     120  050111  051440  EM30:   .ASCIZ  /PIP SET IN RKMR2/
10656  061366  052105  044440  020116
10657  061374  045522  051115  000062
10658  061402  047520  052122  047040  EM31:   .ASCIZ  /PORT NOT AVAILABLE/
10659  061410  052117  040440  040526
10660  061416  046111  041101  042514
10661  061424     000
10662  061425     120  051117  020124  EM32:   .ASCIZ  /PORT AVAILABLE/
10663  061432  053101  044501  040514
10664  061440  046102  000105
10665  061444  052101  047124  051440  EM33:   .ASCIZ  /ATTN SET IN RKASOF/
10666  061452  052105  044440  020116
10667  061460  045522  051501  043117
10668  061466     000
10669  061467     101  052124  020116  EM34:   .ASCIZ  /ATTN CLEARED IN RKASOF/
10670  061474  046103  040505  042522
10671  061502  020104  047111  051040
10672  061510  040513  047523  000106
10673  061516  042503  051122  047040  EM35:   .ASCIZ  /CERR NOT SET IN RKCS1/
10674  061524  052117  051440  052105
10675  061532  044440  020116  045522
10676  061540  051503  000061
10677  061544  054503  020114  042101  EM36:   .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
10678  061552  051104  044440  020116
10679  061560  045522  051115  020063
10680  061566  047516  020124  040523
10681  061574  042515  040440  020123
10682  061602  045522  041504     000
10683  061607     115  046125  044524  EM37:   .ASCIZ  /MULTIPLE ATTENTIONS SEEN/
10684  061614  046120  020105  052101
10685  061622  042524  052116  047511
10686  061630  051516  051440  042505
10687  061636  000116
10688  061640  044524  042515  052517  EM38:   .ASCIZ  /TIMEOUT DID NOT RE-TRIGGER FOR FULL SECOND/
10689  061646  020124  044504  020104
10690  061654  047516  020124  042522
10691  061662  052055  044522  043507
10692  061670  051105  043040  051117
10693  061676  043040  046125  020114
10694  061704  042523  047503  042116
10695  061712     000
10696  061713     103  046131  042040  EM39:   .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
10697  061720  043111  020106  020046
10698  061726  043117  051506  052105
10699  061734  044440  020116  045522
10700  061742  051115  020062  047516
10701  061750  020124  046103  040505
10702  061756  042522  000104
10703  061762  054503  020114  042101  EM40:   .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
10704  061770  051104  044440  020116
10705  061776  045522  051115  020063
```

```
10706  062004  047516  020124  046103
10707  062012  040505  042522  000104
10708  062020  052101  047124  047040  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
10709  062026  052117  041440  042514
10710  062034  051101  042105  044440
10711  062042  020116  045522  051501
10712  062050  043117     000
10713  062053     104  052114  051440  EM63:   .ASCIZ  /DLT SET IN RKCS2/
10714  062060  052105  044440  020116
10715  062066  045522  051503  000062
10716  062074  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERROR/
10717  062102  040505  042504  020122
10718  062110  051105  047522  000122
10719  062116  046101  043511  046516  EM69:   .ASCIZ  /ALIGNMENT CARTRIDGE USED/
10720  062124  047105  020124  040503
10721  062132  052122  044522  043504
10722  062140  020105  051525  042105
10723  062146     000
10724  062147     103  047524  051440  EM73:   .ASCIZ  /CTO SET IN RKCS1/
10725  062154  052105  044440  020116
10726  062162  045522  051503  000061
10727  062170  052122  020132  047516  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
10728  062176  020124  042523  020124
10729  062204  047111  051040  046513
10730  062212  031122     000
10731  062215     116  042105  051440  EM79:   .ASCIZ  /NED SET IN RKCS2/
10732  062222  052105  044440  020116
10733  062230  045522  051503  000062
10734  062236  051127  052111  020105  EM80:   .ASCIZ  /WRITE CHECK ERROR SET IN RKCS2/
10735  062244  044103  041505  020113
10736  062252  051105  047522  020122
10737  062260  042523  020124  047111
10738  062266  051040  041513  031123
10739  062274     000
10740  062275     122  040505  020104  EM82:   .ASCIZ  /READ DATA DID NOT COMPARE WITH WRITE DATA/
10741  062302  040504  040524  042040
10742  062310  042111  047040  052117
10743  062316  041440  046517  040520
10744  062324  042522  053440  052111
10745  062332  020110  051127  052111
10746  062340  020105  040504  040524
10747  062346     000
10748  062347     104  052101  020101  EM83:   .ASCIZ  /DATA CHECK ERROR SET IN RKER/
10749  062354  044103  041505  020113
10750  062362  051105  047522  020122
10751  062370  042523  020124  047111
10752  062376  051040  042513  000122
10753  062404  044127  046111  020105  EM84:   .ASCIZ  /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
10754  062412  040527  052111  047111
10755  062420  020107  047506  020122
10756  062426  047503  052116  020122
10757  062434  042522  042101  020131
10758  062442  051117  040440  052106
10759  062450  051105  041440  047117
10760  062456  051124  051040  040505
10761  062464  054504  051040  041505
```

```
10762  062472  042047     000
10763  062475     122  040505  044504  EM93:   .ASCIZ  /READING WRONG CYLINDER # IN HEADER...MISPOSITION/
10764  062502  043516  053440  047522
10765  062510  043516  041440  046131
10766  062516  047111  042504  020122
10767  062524  020043  047111  044040
10768  062532  040505  042504  027122
10769  062540  027056  044515  050123
10770  062546  051517  052111  047511
10771  062554  000116
10772
10773                          .SBTTL  DATA HEADERS
10774
10775  062556  042524  052123  047040  DH1:    .ASCIZ  /TEST NO.   PC/
10776  062564  027117  020040  041520
10777  062572     000
10778  062573     122  046513  030522  DH2:    .ASCIZ  /RKMR1  RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2/
10779  062600  051011  046513  031122
10780  062606  051011  046513  031522
10781  062614  051011  042513  004522
10782  062622  045522  051504  051011
10783  062630  041513  030523  051011
10784  062636  041513  031123     000
10785  062643     122  053513  004503  DH3:    .ASCIZ  /RKWC   RKBA    RKDA    RKASOF  RKDC    RKECPS  RKECPT/
10786  062650  045522  040502  051011
10787  062656  042113  004501  045522
10788  062664  051501  043117  051011
10789  062672  042113  004503  045522
10790  062700  041505  051520  051011
10791  062706  042513  050103  000124
10792  062714  043101  042524  020122  DH4:    .ASCIZ  /AFTER ATTN REC'D FROM RECAL COMMAND/
10793  062722  052101  047124  051040
10794  062730  041505  042047  043040
10795  062736  047522  020115  042522
10796  062744  040503  020114  047503
10797  062752  046515  047101  000104
10798  062760  043101  042524  020122  DH5:    .ASCIZ  /AFTER SOFTWARE TIMEOUT/
10799  062766  047523  052106  040527
10800  062774  042522  052040  046511
10801  063002  047505  052125     000
10802  063007     106  047522  020115  DH6:    .ASCIZ  /FROM CYL  TO CYL  CYL DIFF/
10803  063014  054503  020114  052040
10804  063022  020117  054503  020114
10805  063030  041440  046131  042040
10806  063036  043111  000106
10807  063042  043101  042524  020122  DH7:    .ASCII  /AFTER SUBSYS CLEAR/
10808  063050  052523  051502  051531
10809  063056  041440  042514  051101
10810  063064  005015  044103  041505          .ASCII  <CR><LF>/CHECK 1. THE PORT B USING DRIVE FLOP (DP1)/
10811  063072  020113  027061  052040
10812  063100  042510  050040  051117
10813  063106  020124  020102  051525
10814  063114  047111  020107  051104
10815  063122  053111  020105  046106
10816  063130  050117  024040  050104
10817  063136  024461
```

```
10818  063140  005015  027062  042040          .ASCIZ  <CR><LF>/2. DRIVE AVAILABLE TO PORT B STATUS (DP2)/
10819  063146  044522  042526  040440
10820  063154  040526  046111  041101
10821  063162  042514  052040  020117
10822  063170  047520  052122  041040
10823  063176  051440  040524  052524
10824  063204  020123  042050  031120
10825  063212  000051
10826  063214  042524  052123  047040  DH8:    .ASCIZ  /TEST NO.       TRAP PC/
10827  063222  027117  052011  040522
10828  063230  020120  041520     000
10829  063235     101  052106  051105  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
10830  063242  051440  040524  052122
10331  063250  051440  044520  042116
10832  063256  042514  041440  046517
10833  063264  040515  042116  051040
10834  063272  041505  042047  041040
10835  063300  020131  051104  053111
10836  063306  000105
10837  063310  052101  042440  042116  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
10838  063316  047440  020106  042510
10839  063324  042101  046040  040517
10840  063332  044504  043516     000
10841  063337     105  050130  041505  DH11:   .ASCIZ  /EXPECTED       WAS/
10842  063344  042524  004504  040527
10843  063352  000123
10844  063354  043101  042524  020122  DH12:   .ASCII  /AFTER SEEK CMD/
10845  063362  042523  045505  041440
10846  063370  042115
10847  063372  005015  044103  041505          .ASCIZ  <CR><LF>/CHECK STAT ADD SIGNAL BEFORE & AFTER MUX/
10848  063400  020113  052123  052101
10849  063406  040440  042104  051440
10850  063414  043511  040516  020114
10851  063422  042502  047506  042522
10852  063430  023040  040440  052106
10853  063436  051105  046440  054125
10854  063444     000
10855  063445     117  020116  042523  DH13:   .ASCIZ  /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
10856  063452  052103  051117  020123
10857  063460  030061  020054  031061
10858  063466  020054  032061  020054
10859  063474  033061  020054  034061
10860  063502  047440  020122  030062
10861  063510  041440  046131  032040
10862  063516  030061  052040  040522
10863  063524  045503  031040     000
10864  063531     102  043105  051117  DH14:   .ASCIZ  /BEFORE TIMEOUT OR RELEASE/
10865  063536  020105  044524  042515
10866  063544  052517  020124  051117
10867  063552  051040  046105  040505
10868  063560  042523     000
10869  063563     127  044510  042514  DH15:   .ASCIZ  /WHILE PORT UNAVAILABLE/
10870  063570  050040  051117  020124
10871  063576  047125  053101  044501
10872  063604  040514  046102  000105
10873  063612  047524  044440  042116  DH16:   .ASCII  /TO INDICATE THAT REQUESTING PORT CAN SEIZE/
```

```
10874  063620  041511  052101  020105
10875  063626  044124  052101  051040
10876  063634  050505  042525  052123
10877  063642  047111  020107  047520
10878  063650  052122  041440  047101
10879  063656  051440  044505  042532
10880  063664  005015  044103  041505          .ASCIZ  <CR><LF>/CHECK PORT REQ. FLOPS & 1 SEC SAFETY TIMER/
10881  063672  020113  047520  052122
10882  063700  051040  050505  020056
10883  063706  046106  050117  020123
10884  063714  020046  020061  042523
10885  063722  020103  040523  042506
10886  063730  054524  052040  046511
10887  063736  051105     000
10888  063741     101  052106  051105  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
10889  063746  051040  041505  046101
10890  063754  041440  046517  040515
10891  063762  042116     000
10892  063765     101  052106  051105  DH18:   .ASCIZ  /AFTER UNLOAD COMMAND/
10893  063772  052440  046116  040517
10894  064000  020104  047503  046515
10895  064006  047101  000104
10896  064012  043101  042524  020122  DH19:   .ASCIZ  /AFTER PACK COMMAND/
10897  064020  040520  045503  041440
10898  064026  046517  040515  042116
10899  064034     000
10900  064035     101  052106  051105  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
10901  064042  051440  046105  041505
10902  064050  020124  051104  053111
10903  064056  020105  047503  046515
10904  064064  047101  000104
10905  064070  043101  042524  020122  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
10906  064076  052523  051502  051531
10907  064104  042524  020115  046103
10908  064112  040505  000122
10909  064116  043101  042524  020122  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
10910  064124  051104  053111  020105
10911  064132  046103  040505  020122
10912  064140  047503  046515  047101
10913  064146  000104
10914  064150  044527  044124  052517  DH23:   .ASCIZ  /WITHOUT REQUEST PENDING/
10915  064156  020124  042522  052521
10916  064164  051505  020124  042520
10917  064172  042116  047111  000107
10918  064200  054502  042040  044522  DH24:   .ASCIZ  /BY DRIVE CLEAR COMMAND TO OTHER PORT/
10919  064206  042526  041440  042514
10920  064214  051101  041440  046517
10921  064222  040515  042116  052040
10922  064230  020117  052117  042510
10923  064236  020122  047520  052122
10924  064244     000
10925  064245     101  052106  051105  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
10926  064252  051440  042505  020113
10927  064260  047503  046515  047101
10928  064266  000104
10929  064270  043101  042524  020122  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
```

```
10930  064276  042522  042101  042040
10931  064304  052101  020101  047503
10932  064312  046515  047101  000104
10933  064320  043101  042524  020122  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
10934  064326  051127  052111  020105
10935  064334  040504  040524  041440
10936  064342  046517  040515  042116
10937  064350     000
10938  064351     011  054105  042520  DH28:   .ASCIZ  /        EXPECT/
10939  064356  052103     000
10940  064361     011  041501  052524  DH29:   .ASCIZ  /        ACTUAL/
10941  064366  046101     000
10942  064371     101  052106  051105  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
10943  064376  051040  040505  020104
10944  064404  042510  042101  051105
10945  064412  041440  046517  040515
10946  064420  042116     000
10947  064423     101  004460  030102  DH31:   .ASCIZ  /A0      B0      A1      B1      A2      B2      B3/
10948  064430  040411  004461  030502
10949  064436  040411  004462  031102
10950  064444  041011  000063
10951  064450  043101  042524  020122  DH32:   .ASCIZ  /AFTER WRITE CHECK COMMAND/
10952  064456  051127  052111  020105
10953  064464  044103  041505  020113
10954  064472  047503  046515  047101
10955  064500  000104
10956  064502  043101  042524  020122  DH35:    ASCIZ  /AFTER RELEASE ISSUED/
10957  064510  042522  042514  051501
10958  064516  020105  051511  052523
10959  064524  042105     000
10960  064527     101  052106  051105  DH36:   .ASCIZ  /AFTER SELECT DRIVE COMMAND & DRIVE NOT AVAILABLE/
10961  064534  051440  046105  041505
10962  064542  020124  051104  053111
10963  064550  020105  047503  046515
10964  064556  047101  020104  020046
10965  064564  051104  053111  020105
10966  064572  047516  020124  053101
10967  064600  044501  040514  046102
10968  064606  000105
10969  064610  043101  042524  020122  DH37:   .ASCIZ  /AFTER RELEASE ISSUED & PORT REQUEST PENDING/
10970  064616  042522  042514  051501
10971  064624  020105  051511  052523
10972  064632  042105  023040  050040
10973  064640  051117  020124  042522
10974  064646  052521  051505  020124
10975  064654  042520  042116  047111
10976  064662  000107
10977  064664  043101  042524  020122  DH38:   .ASCIZ  /AFTER DELAY FROM SEEK & IMMEDIATE RELEASE/
10978  064672  042504  040514  020131
10979  064700  051106  046517  051440
10980  064706  042505  020113  020046
10981  064714  046511  042515  044504
10982  064722  052101  020105  042522
10983  064730  042514  051501  000105
10984  064736  043101  042524  020122  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
10985  064744  051127  052111  020105
```

```
10986  064752  042510  042101  051105
10987  064760  041440  046517  040515
10988  064766  042116     000
10989  064771     127  051117  021504  DH40:   .ASCIZ  /WORD#  HEADER WAS  SHOULD BE/
10990  064776  044011  040505  042504
10991  065004  020122  040527  020123
10992  065012  051440  047510  046125
10993  065020  020104  042502     000
10994  065025     104  051125  047111  DH41:   .ASCII  /DURING RECAL COMMAND/
10995  065032  020107  042522  040503
10996  065040  020114  047503  046515
10997  065046  047101     104
10998  065051     015  041412  042510          .ASCIZ  <CR><LF>/CHECK BUFF1 SIGNAL BEFORE & AFTER MUX/
10999  065056  045503  041040  043125
11000  065064  030506  051440  043511
11001  065072  040516  020114  042502
11002  065100  047506  042522  023040
11003  065106  040440  052106  051105
11004  065114  046440  054125     000
11005  065121     117  020116  042523  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
11006  065126  052103  051117  020123
11007  065134  026060  026062  026064
11008  065142  020066  051117  034040
11009  065150  020040  054503  020114
11010  065156  030464  020060  051124
11011  065164  041501  020113  000062
11012  065172  043101  042524  020122  DH43:   .ASCIZ  /AFTER RE-SEIZE  PORT MID-WAY IN NORMAL TIMEOUT/
11013  065200  042522  051455  044505
11014  065206  042532  020040  047520
11015  065214  052122  046440  042111
11016  065222  053455  054501  044440
11017  065230  020116  047516  046522
11018  065236  046101  052040  046511
11019  065244  047505  052125     000
11020  065251     106  051117  040515  DH44:   .ASCIZ  /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
11021  065256  020124  020046  046101
11022  065264  020114  042522  042101
11023  065272  053455  044522  042524
11024  065300  052040  051505  051524
11025  065306  053440  046111  020114
11026  065314  042502  041040  050131
11027  065322  051501  042523  000104
11028  065330  042502  047506  042522  DH45:   .ASCIZ  /BEFORE RELEASE WHILE HEADS UNLOADED/
11029  065336  051040  046105  040505
11030  065344  042523  053440  044510
11031  065352  042514  044040  040505
11032  065360  051504  052440  046116
11033  065366  040517  042504  000104
11034  065374  051515  020107  020101  DH49:   .ASCIZ  /MSG A & B IN RKMR2 & RKMR3 RESP.  ARE INVALID/
11035  065402  020046  020102  047111
11036  065410  051040  046513  031122
11037  065416  023040  051040  046513
11038  065424  031522  051040  051505
11039  065432  027120  040440  042522
11040  065440  044440  053116  046101
11041  065446  042111     000
```

```
11042  065451  000101  052106  051105  DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
11043  065456  051440  042505  020113
11044  065464  047524  051440  046105
11045  065472  020106  047503  046515
11046  065500  047101  000104
11047  065504  054503  020114  004443  DH56:   .ASCIZ  /CYL #  HEADER WORD 0/
11048  065512  042510  042101  051105
11049  065520  053440  051117  020104
11050  065526  000060
11051
11052                                  .SBTTL  ERROR OUTPUT DATA
11053
11054                                          .EVEN
11055  065530  001214  001116          DT1:    $TESTN,$ERRPC
11056  065534  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11057  065542  005330  005326  005314
11058  065550  005316
11059  065552  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11060  065560  005332  005334  005346
11061  065566  005350
11062  065570  001214  001334          DT3:    $TESTN,TRAPPC
11063  065574  001214  001116  001344  DT4:    $TESTN,$ERRPC,FRCYL,TOCYL,CALDIF
11064  065602  001346  001354
11065  065606  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11066  065614  005330  005326  005314
11067  065622  005316
11068  065624  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11069  065632  005332  005334  005346
11070  065640  005350
11071  065642  001214  001116  001416  DT6:    $TESTN,$ERRPC,WD2,WD1
11072  065650  001414
11073  065652  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11074  065660  005330  005326  005314
11075  065666  005316
11076  065670  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11077  065676  005332  005334  005346
11078  065704  005350
11079  065706  001214  001116  001432  DT7:    $TESTN,$ERRPC,WDCNT,HDWD,TEMP1
11080  065714  001450  005352
11081  065720  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11082  065726  005330  005326  005314
11083  065734  005316
11084  065736  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11085  065744  005332  005334  005346
11086  065752  005350
11087  065754  001214  001116  001346  DT8:    $TESTN,$ERRPC,TOCYL,FRCYL,CALDIF
11088  065762  001344  001354
11089  065766  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11090  065774  005330  005326  005314
11091  066002  005316
11092  066004  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11093  066012  005332  005334  005346
11094  066020  005350
11095  066022  001214  001116  001346  DT9:    $TESTN,$ERRPC,TOCYL,RHTAB
11096  066030  001666
11097  066032  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
```

```
11098  066040  005330  005326  005314
11099  066046  005316
11100  066050  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11101  066056  005332  005334  005346
11102  066064  005350
11103  066066  001214  001116  005404  DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
11104  066074  005406  005410  005412
11105  066102  005364  005366  005370
11106  066110  005372
11107  066112  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11108  066120  005330  005326  005314
11109  066126  005316
11110  066130  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11111  066136  005332  005334  005346
11112  066144  005350
11113  066146  001214  001116  005404  DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
11114  066154  005406  005410  005412
11115  066162  005414  005416
11116  066166  005364  005366  005370          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
11117  066174  005372  005374  005376
11118  066202  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11119  066210  005330  005326  005314
11120  066216  005316
11121  066220  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11122  066226  005332  005334  005346
11123  066234  005350
11124  066236  001214  001116  005404  DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
11125  066244  005406  005410  005412
11126  066252  005414  005416  005422
11127  066260  005364  005366  005370          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
11128  066266  005372  005374  005376
11129  066274  005402
11130  066276  005340  005342  005344          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11131  066304  005330  005326  005314
11132  066312  005316
11133  066314  005320  005322  005324          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11134  066322  005332  005334  005346
11135  066330  005350
11136
11137                                  .SBTTL  ERROR DATA FORMATS
11138
11139  066332  000003                  DF1:    3
11140  066334     002      000                 .BYTE 2,0
11141  066336  062573                          DH2
11142  066340     007      000                 .BYTE 7,0
11143  066342  062643                          DH3
11144  066344     007      000                 .BYTE   7,0
11145
11146  066346  000001                  DF2:    1
11147  066350     002      000                 .BYTE   2,0
11148
11149  066352  000005                  DF3:    5
11150  066354     000      000                 .BYTE   0,0
11151  066356  062556                          DH1
11152  066360     002      000                 .BYTE   2,0
11153  066362  063337                          DH11
```

```
11154  066364    002     000              .BYTE   2,0
11155  066366  062573                     DH2
11156  066370    007     000              .BYTE   7,0
11157  066372  062643                     DH3
11158  066374    007     000              .BYTE   7,0
11159
11160  066376  000003             DF4:    3
11161  066400    002     000              .BYTE   2,0
11162  066402  062573                     DH2
11163  066404    007     000              .BYTE   7,0
11164  066406  062643                     DH3
11165  066410    007     000              .BYTE   7,0
11166
11167  066412  000005             DF5:    5
11168  066414    000     000              .BYTE   0,0
11169  066416  065374                     DH49
11170  066420    000     000              .BYTE   0,0
11171  066422  062556                     DH1
11172  066424    002     000              .BYTE   2,0
11173  066426  062573                     DH2
11174  066430    007     000              .BYTE   7,0
11175  066432  062643                     DH3
11176  066434    007     000              .BYTE   7,0
11177
11178  066436  000005             DF6:    5
11179  066440    000     000              .BYTE 0,0
11180  066442  062556                     DH1
11181  066444    002     000              .BYTE 2,0
11182  066446  063007                     DH6
11183  066450    003     000              .BYTE 3,0
11184  066452  062573                     DH2
11185  066454    007     000              .BYTE   7,0
11186  066456  062643                     DH3
11187  066460    007     000              .BYTE   7,0
11188
11189
11190  066462  000004             DF10:   4
11191  066464    000     000              .BYTE 0,0
11192  066466  062556                     DH1
11193  066470    002     000              .BYTE 2,0
11194  066472  062573                     DH2
11195  066474    007     000              .BYTE 7,0
11196  066476  062643                     DH3
11197  066500    007     000              .BYTE   7,0
11198
11199  066502  000004             DF14:   4
11200  066504    002     000              .BYTE   2,0
11201  066506  064771                     DH40
11202  066510    003     000              .BYTE   3,0
11203  066512  062573                     DH2
11204  066514    007     000              .BYTE   7,0
11205  066516  062643                     DH3
11206  066520    007     000              .BYTE   7,0
11207
11208
11209  066522  000004             DF15:   4
```

```
11210  066524  000     000              .BYTE   0,0
11211  066526  062556                   DH1
11212  066530  002     000              .BYTE   2,0
11213  066532  062573                   DH2
11214  066534  007     000              .BYTE   7,0
11215  066536  062643                   DH3
11216  066540  007     000              .BYTE   7,0
11217
11218  066542  000005          DF17:    5
11219  066544  000     000              .BYTE   0,0
11220  066546  065251                   DH44
11221  066550  000     000              .BYTE   0,0
11222  066552  062556                   DH1
11223  066554  002     000              .BYTE   2,0
11224  066556  062573                   DH2
11225  066560  007     000              .BYTE   7,0
11226  066562  062643                   DH3
11227  066564  007     000              .BYTE   7,0
11228  066566  000005          DF20:    5
11229  066570  000     000              .BYTE   0,0
11230  066572  062556                   DH1
11231  066574  002     000              .BYTE   2,0
11232  066576  065504                   DH56
11233  066600  002     000              .BYTE   2,0
11234  066602  062573                   DH2
11235  066604  007     000              .BYTE   7,0
11236  066606  062643                   DH3
11237  066610  007     000              .BYTE   7,0
11238
11239  066612  000007          DF21:    7
11240  066614  000     000              .BYTE   0,0
11241  066616  062556                   DH1
11242  066620  002     000              .BYTE   2,0
11243  066622  064351                   DH28
11244  066624  000     000              .BYTE   0,0
11245  066626  064423                   DH31
11246  066630  004     000              .BYTE   4,0
11247  066632  064361                   DH29
11248  066634  004     000              .BYTE   4,0
11249  066636  062573                   DH2
11250  066640  007     000              .BYTE   7,0
11251  066642  062643                   DH3
11252  066644  007     000              .BYTE   7,0
11253
11254  066646  000007          DF22:    7
11255  066650  000     000              .BYTE   0,0
11256  066652  062556                   DH1
11257  066654  002     000              .BYTE   2,0
11258  066656  064351                   DH28
11259  066660  000     000              .BYTE   0,0
11260  066662  064423                   DH31
11261  066664  006     000              .BYTE   6,0
11262  066666  064361                   DH29
11263  066670  006     000              .BYTE   6,0
11264  066672  062573                   DH2
11265  066674  007     000              .BYTE   7,0
```

```
11266  066676  062643                          DH3
11267  066700     007     000                  .BYTE   7,0
11268
11269  066702  000007                       DF23:  7
11270  066704     000     000                  .BYTE   0,0
11271  066706  062556                          DH1
11272  066710     002     000                  .BYTE   2,0
11273  066712  064351                          DH28
11274  066714     000     000                  .BYTE   0,0
11275  066716  064423                          DH31
11276  066720     007     000                  .BYTE   7,0
11277  066722  064361                          DH29
11278  066724     007     000                  .BYTE   7,0
11279  066726  062573                          DH2
11280  066730     007     000                  .BYTE   7,0
11281  066732  062643                          DH3
11282  066734     007     000                  .BYTE   7,0
11283                                  .EVEN
11284                                  ;;*****************************************************************
11285                                  .SBTTL  TYPE ERROR ROUTINE
11286                                  ;*ENTRY JSR PC,TYP ERR
11287                                  ;*RETURN RTS PC
11288                                  ;*
11289                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
11290                                  ;*ERROR IS TO BE REPORTED.  IT THEN USES THE "ERROR TABLE" ($ERRTB)
11291                                  ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
11292                                  ;*THE ERROR.
11293                                  ;;*****************************************************************
11294  066736  104413                  TYPERR: SAVREG
11295  066740  104401  056567                  TYPE    ,MSG19          ;PORT A OR B
11296  066744  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
11297  066750  042700  177400                  BIC     #177400,R0      ;CLEAR SIGN EXTENSION
11298  066754  005300                          DEC     R0              ;FORM INDEX FOR ERROR TABLE
11299  066756  006300                          ASL     R0
11300  066760  006300                          ASL     R0
11301  066762  006300                          ASL     R0
```

```
11302  066764  062700  005470          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
11303  066770  012037  067004                  MOV     (R0)+,2$        ;GET EM POINTER
11304  066774  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
11305  066776  104401  001205                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
11306  067002  104401                          TYPE                    ;TYPE ERROR MESSAGE (EM)
11307  067004  000000          2$:     .WORD   0                       ;EM POINTER GOES HERE
11308  067006  012037  067022  3$:     MOV     (R0)+,4$                ;GET DH POINTER
11309  067012  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
11310  067014  104401  001205                  TYPE    ,$CRLF          ;TYPE CR-LF
11311  067020  104401                          TYPE                    ;TYPE DATA HEADER
11312  067022  000000          4$:     .WORD   0                       ;DH POINTER GOES HERE
11313  067024  012001          5$:     MOV     (R0)+,R1                ;GET DT POINTER
11314  067026  001455                          BEQ     20$             ;BRANCH IF THERE ARE NONE
11315  067030  005004                          CLR     R4              ;SET INDENT SWITCH
11316  067032  012000                          MOV     (R0)+,R0        ;GET DF POINTER
11317  067034  012002                          MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
11318  067036  001446                          BEQ     17$             ;DH NUM IS 0-BRANCH
11319  067040  005104                          COM     R4              ;NO INDENT
11320  067042  104401  001205                  TYPE    ,$CRLF
11321  067046  112003          10$:    MOVB    (R0)+,R3                ;GET & STORE NUMBER OF DATA WORDS
11322  067050  105720                          TSTB    (R0)+           ;BUMP PAST FORMAT WORD
11323  067052  005703                          TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
11324  067054  001407                          BEQ     14$             ;NO - SKIP DATA PRINT
11325  067056  013146          11$:    MOV     @(R1)+,-(SP)            ;PUT FIRST DATA WORD ON STACK
11326  067060  104402                          TYPOC                   ;TYPE IT
11327  067062  005303                          DEC     R3              ;MORE DATA WORDS
11328  067064  001403                          BEQ     14$             ;NO-BRANCH
```

```
11329  067066  104401  067216              TYPE    ,SPACE2          ;TYPE SEPARATORS
11330  067072  000771                      BR      11$              ;LOOP
11331  067074  005302              14$:    DEC     R2               ;MORE DH'S?
11332  067076  003431                      BLE     20$              ;NO-BRANCH
11333  067100  104401  001205              TYPE    ,$CRLF
11334  067104  005760  000002              TST     2(R0)            ;ONLY A DH IN THIS REQUEST?
11335  067110  001404                      BEQ     15$              ;YES-BRANCH BYPASS INDENT
11336  067112  005104                      COM     R4               ;INDENT?
11337  067114  001002                      BNE     15$              ;NO-BRANCH
11338  067116  104401  067216              TYPE    ,SPACE2          ;YES-TYPE SPACES
11339  067122  012037  067130      15$:    MOV     (R0)+,16$        ;GET NEXT DH POINTER
11340  067126  104401                      TYPE                     ;TYPE DH
11341  067130  000000              16$:    .WORD   0                ;DH POINTER GOES HERE
11342  067132  105710                      TSTB    (R0)             ;TYPE A DT?
11343  067134  001003                      BNE     21$              ;YES-BRANCH
11344  067136  062700  000002              ADD     #2,R0            ;INCREMENT DF POINTER
11345  067142  000754                      BR      14$              ;SEE IF END OF DF BLOCK
11346  067144  104401  001205      21$:    TYPE    ,$CRLF
11347  067150  005704                      TST     R4               ;INDENT?
11348  067152  001335                      BNE     10$              ;NO-BRANCH
11349  067154  104401  067216      17$:    TYPE    ,SPACE2          ;YES-TYPE SPACES
11350  067160  000732                      BR      10$              ;LOOP
11351  067162  104414              20$:    RESREG
11352
11353  067164  032777  010000  111746      BIT     #SW12,@SWR       ;SEE IF ABORT DRV AFTER 20 ERRORS
11354  067172  001410                      BEQ     25$              ;BR IF NO
11355  067174  023727  001103  000024      CMP     $ERFLG,#20.      ;ELSE SEE IF HAVE 20 ERRORS
11356  067202  001004                      BNE     25$              ;BR IF NO
11357  067204  012706  001100              MOV     #STACK,SP        ;ELSE RESTORE STACK PTR
11358  067210  000137  042644              JMP     $EOP             ;AND GO TO NEXT DRV
11359
11360  067214  000207              25$:    RTS     PC
11361  067216  020040     000      SPACE2: .ASCIZ/  /               ;2 SPACES
11362                               ;  ODT-11 -- V005A
11363
11364                               ; DEC-11-UODPA-A-LA
11365
11366                               ; COPYRIGHT 1969,1970,1972
11367                               ; DIGITAL EQUIPMENT CORPORATION
11368                               ; MAYNARD, MASSACHUSETTS 01754
11369                                       .ENABL  ABS,AMA
11370          067222                       .EVEN
11371          067302                       .=.+60
11372          000000               R0      =       %0       ; REGISTER
11373          000001               R1      =       %1       ;  NAMING
11374          000002               R2      =       %2       ;   CONVENTIONS
11375          000003               R3      =       %3
11376          000004               R4      =       %4
11377          000005               R5      =       %5
11378          000006               SP      =       %6
11379          000007               PC      =       %7
11380          177776               ST      =       177776           ;STATUS REGISTER
11381                               ;
11382          000014               0.TVEC  =       14               ;TRT VECTOR LOCATION
11383          000340               0.STM   =       340              ;PRIORITY MASK - STATUS REGISTER
11384          000020               0.TBT   =       20               ;T-BIT MASK - STATUS REGISTER
```

```
11385           000003                  TRT     =       000003          ;TRT INSTRUCTION
11386           000006                  RTT     =       000006          ;RTT INSTRUCTION
11387                                   ;
11388                                   ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
11389                                   ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
11390                                   ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
11391                                   ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
11392                                   ;
11393
11394
11395
11396           177562                  0.RDB   =       177562  ;R DATA BUFFER
11397           177560                  0.RCSR  =       177560  ;R C/SR
11398           177566                  0.TDB   =       177566  ;T DATA BUFFER
11399           177564                  0.TCSR  =       177564  ;T C/SR
11400                                   ;
11401                                   ;
11402                                   ; INITIALIZE ODT
11403                                   ;  USE 0.ODT FOR A NORMAL ENTRY
11404                                   ;  USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
11405                                   ;  USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
11406                                   ;
11407  067302   000413                  0.ODT:  BR      0.STRT          ;NORMAL ENTRY
11408  067304   000417                          BR      0.RST           ;RESTART
11409  067306   013737   177776  067262  0.ENTR: MOV     ST,0.UST        ;RE-ENTER  --  SAVE STATUS
11410  067314   013737   000016  177776          MOV     0.TVEC+2,ST     ;SET UP LOCAL STATUS
11411  067322   010737   067260                  MOV     PC,0.UPC        ;FAKE THE PC
11412  067326   000137   070460                  JMP     0.BK1
11413                                   ;
11414  067332   012706   067242          0.STRT: MOV     #0.URO,SP       ;SET UP STACK
11415  067336   010637   067256                  MOV     SP,0.USP        ;FAKE THE SAVED STACK
11416  067342   000414                          BR      0.RST1          ;CLEAR BREAKPOINT TABLES
11417  067344   004037   070666          0.RST:  JSR     0,0.SVR         ;SAVE REGISTERS
11418  067350   013777   067300  177716          MOV     0.UIN,@0.ADR1   ;REMOVE THE BREAKPOINT
11419  067356   113704   067264                  MOVB    0.PRI,R4        ;GET ODT PRIORITY
11420  067362   106004                          RORB    R4              ;SHIFT
11421  067364   106004                          RORB    R4              ;  INTO
11422  067366   106004                          RORB    R4              ;    POSITION
11423  067370   110437   177776                  MOVB    R4,ST           ;STORE IN STATUS
11424  067374   000127                  0.RST1: JMP     (PC)+
11425  067376   000403                          BR      0.45
11426  067400   012737   000002  070370          MOV     #RTI,0.RTIT     ;SET TO RTI IF 11/20 OR /05
11427  067406   105037   071307          0.45:   CLRB    0.P             ;DISALLOW PROCEED
11428  067412   012737   000340  000016          MOV     #0.STM,0.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
11429  067420   012737   070450  000014          MOV     #0.BRK,0.TVEC   ;PC TO TRT VECTOR
11430  067426   000447                          BR      0.RALL          ;CLEAR BREAKPOINT TABLES
11431                                   ;
11432                                   ; SPECIAL NAME HANDLER
11433                                   ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URO
11434                                   ;
11435  067430   004537   071110          0.REGT: JSR     5,0.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
11436  067434   012704   071333                  MOV     #0.TL,R4        ;TABLE START ADDRESS
11437  067440   120024                  0.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
11438  067442   001413                          BEQ     0.SP            ;JUMP IF YES
11439  067444   022704   071341                  CMP     #0.TL+0.LG,R4   ;IS THE SEARCH DONE?
11440  067450   101373                          BHI     0.RSP           ;BRANCH IF NOT
```

```
11441  067452  042700  177770            BIC     #177770,R0      ;MASK OFF OCTAL
11442  067456  010004                    MOV     R0,R4
11443  067460  006304            0.SP1:  ASL     R4
11444  067462  062704  067242            ADD     #0.URO,R4       ;GENERATE ADDRESS
11445  067466  005202                    INC     R2              ;SET FOUND FLAG
11446  067470  000444                    BR      0.SCAN          ;GO FIND NEXT CHARACTER
11447  067472  162704  071324    0.SP:   SUB     #0.TL-7,R4      ;CORRECT CONSTANT
11448  067476  000770                    BR      0.SP1
11449                                    ;
11450                                    ; _ HANDLER - OPEN INDEXED ON THE PC
11451                                    ;
11452  067500  004737  071234    0.ORPC: JSR     PC,0.TCLS
11453  067504  010502                    MOV     R5,R2           ;CURRENT ADDRESS IN R2
11454  067506  061202                    ADD     @R2,R2          ;COMPUTE
11455  067510  006202                    ASR     R2              ;MOVE ONE BIT TO CARRY
11456  067512  103421                    BCS     0.ERR           ;ERROR IF ODD NUMBER
11457  067514  006302                    ASL     R2              ;RESTORE WORD
11458  067516  005722                    TST     (R2)+           ; AND INCREMENT BY TWO
11459  067520  010205                    MOV     R2,R5           ;UPDATE CAD
11460  067522  000137  067774            JMP     0.OP2           ;GO FINISH UP
11461                                    ;
11462                                    ; B HANDLER - SET AND REMOVE BREAKPOINTS
11463                                    ;
11464  067526  005702            0.BKPT: TST     R2              ;IF NO NUMBER TYPED
11465  067530  001406                    BEQ     0.RALL          ;  REMOVE BREAKPOINT
11466  067532  006204                    ASR     R4              ;CHECK IF ODD
11467  067534  103410                    BCS     0.ERR           ;JUMP IF ODD
11468  067536  006304                    ASL     R4              ;RESTORE ONE BIT
11469  067540  010437  067274            MOV     R4,0.ADR1       ;SET A BREAKPOINT
11470  067544  000412                    BR      0.DCD
11471  067546  012737  071350  067274  0.RALL: MOV     #0.TRTC,0.ADR1  ;CLEAR BREAKPOINT
11472  067554  000406                    BR      0.DCD
11473                                    ;
11474                                    ; COMMAND DECODER - ODT11
11475                                    ;
11476                                    ;   REGISTERS R0-R4 MAY BE USED,
11477                                    ;    REGISTER R5 WILL BE CONSIDERED SAFE
11478                                    ;
11479  067556  052705  000001    0.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
11480  067562  012700  000077            MOV     #'?,R0          ;  ? TO BE TYPED
11481  067566  004537  071166            JSR     5,0.FTYP        ; OUTPUT ?
11482  067572  004537  071266    0.DCD:  JSR     5,0.CRLS        ;TYPE  <CR><LF>*
11483  067576  005004            0.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
11484  067600  005002                    CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
11485  067602  004537  071110    0.SCAN: JSR     5,0.GET         ;GET A CHAR, RETURN IN R0
11486  067606  022700  000060            CMP     #'0,R0          ;COMPARE WITH ASCII 0
11487  067612  101013                    BHI     0.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
11488  067614  022700  000067            CMP     #'7,R0          ;COMPARE WITH ASCII 7
11489  067620  103410                    BLO     0.CLGL          ;CHECK LEGALITY IF NOT OCTAL
11490  067622  042700  177770            BIC     #177770,R0      ;CONVERT TO BCD
11491  067626  006304                    ASL     R4              ; MAKE ROOM
11492  067630  006304                    ASL     R4              ;   IN
11493  067632  006304                    ASL     R4              ;     R4
11494  067634  060004                    ADD     R0,R4           ;PACK THREE BITS IN R4
11495  067636  005202                    INC     R2              ;R2 HAS NUMERIC FLAG
11496  067640  000760                    BR      0.SCAN          ;    AND TRY AGAIN
```

```
11497  067642  005001                  0.CLGL: CLR     R1                  ;CLEAR INDEX
11498  067644  120061  071317          0.LGL1: CMPB    R0,0.LGCH(R1)       ;DO THE CODES MATCH?
11499  067650  001405                          BEQ     0.LGL2              ;JUMP IF YES
11500  067652  005201                          INC     R1                  ; SET INDEX FOR NEXT SEARCH
11501  067654  020127  000014                  CMP     R1,#0.CLGT          ;IS THE SEARCH DONE?
11502  067660  103336                          BHIS    0.ERR               ;    OOPS!
11503  067662  000770                          BR      0.LGL1              ;RE-LOOP
11504  067664  006301                  0.LGL2: ASL     R1                  ;MULTIPLY BY TWO
11505  067666  000171  067672                  JMP     @0.LGDR(R1)         ;GO TO PROPER ROUTINE
11506                                  ;
11507  067672  067722                  0.LGDR: 0.WRD   ;  /    OPEN WORD
11508  067674  067754                          0.CRET  ;  CARRIAGE RETURN    CLOSE
11509  067676  067430                          0.REGT  ;  $  REGISTER OPS
11510  067700  070264                          0.GO    ;  G  GO TO ADDRESS K
11511  067702  067766                          0.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
11512  067704  067500                          0.ORPC  ;  _  OPEN RELATED, INDEX - PC
11513  067706  070020                          0.BACK  ;     OPEN PREVIOUS
11514
11515  067710  070030                          0.OFST  ;  O  OFFSET
11516  067712  070106                          0.WSCH  ;  W  SEARCH WORD
11517  067714  070102                          0.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
11518  067716  067526                          0.BKPT  ;  B  BREAKPOINTS
11519  067720  070372                          0.PROC  ;  P  PROCEED
11520          000030                  0.LGL   =       .-0.LGDR            ;LGL MUST EQUAL 2X CHLGT ALWAYS
11521                                  ;
11522                                  ; PROCESS / - OPEN WORD
11523                                  ;
11524  067722  005702                  0.WRD:  TST     R2                  ;GET VALUE IF R2 IS NON-ZERO
11525  067724  001410                          BEQ     0.WRDA              ;SKIP OTHERWISE
11526  067726  010405                          MOV     R4,R5               ;    PUT VALUE IN CAD
11527  067730  006205                  0.WRD1: ASR     R5                  ;MOVE ONE BIT TO CARRY
11528  067732  103711                  0.ERR2: BCS     0.ERR               ;JUMP IF ODD ADDRESS
11529  067734  006305                          ASL     R5                  ;RESTORE THE CARRY BIT
11530  067736  011500                          MOV     @R5,R0              ;GET CONTENTS OF WORD
11531  067740  004537  071024                  JSR     5,0.CADV            ;GO GET AND TYPE OUT @CAD
11532  067744  000714                          BR      0.DCD1              ;GO BACK TO DECODER
11533  067746  042705  000001          0.WRDA: BIC     #1,R5               ;CLEAR CLOSED BIT
11534  067752  000766                          BR      0.WRD1              ;GO BACK TO MAIN-LINE
11535                                  ;
11536                                  ; PROCESS CARRIAGE RETURN
11537                                  ;
11538  067754  004737  071234          0.CRET: JSR     PC,0.TCLS           ;CLOSE LOCATION
11539  067760  052705  000001                  BIS     #1,R5               ;CLOSE EVERYTHING
11540  067764  000702                          BR      0.DCD               ;RETURN TO DECODER
11541                                  ;
11542                                  ; PROCESS <LF>, OPEN NEXT WORD
11543                                  ;
11544  067766  004737  071234          0.OP1:  JSR     PC,0.TCLS           ;CLOSE PRESENT CELL
11545  067772  005725                          TST     (R5)+               ;GENERATE NEW ADDRESS
11546  067774  004537  071260          0.OP2:  JSR     5,0.CRLF            ;<CR><LF>
11547  070000  010500                          MOV     R5,R0               ;NUMBER TO TYPE
11548  070002  004537  071024                  JSR     5,0.CADV            ;  TYPE OUT ADDRESS
11549  070006  012700  000057                  MOV     #'/,R0              ;TYPE A /
11550  070012  004537  071166                  JSR     5,0.FTYP
11551  070016  000744                          BR      0.WRD1              ;GO PROCESS IT
11552                                  ;
```

```
11553                                   ; PROCESS , OPEN PREVIOUS WORD
11554                                   ;
11555  070020  004737  071234          0.BACK: JSR     PC,0.TCLS
11556  070024  005745                          TST     -(R5)           ;GENERATE NEW ADDRESS
11557  070026  000762                          BR      0.OP2           ;GO DO THE REST
11558                                   ;
11559                                   ; PROCESS 0, COMPUTE OFFSET
11560                                   ;
11561  070030  006205                  0.OFST: ASR     R5              ;GET LOW ORDER BIT
11562  070032  103737                          BCS     0.ERR2          ;ERROR IF CLOSED
11563  070034  006305                          ASL     R5              ;RESTORE WORD
11564  070036  012700  000040                  MOV     #' ,R0          ;TYPE ONE BLANK
11565  070042  004537  071166                  JSR     5,0.FTYP        ; AS A SEPARATOR
11566  070046  160504                          SUB     R5,R4           ;COMPUTE
11567  070050  005304                          DEC     R4
11568  070052  005304                          DEC     R4              ;    16 BIT OFFSET
11569  070054  010400                          MOV     R4,R0           ;TYPE A
11570  070056  010402                          MOV     R4,R2           ;SAVE R4
11571  070060  004537  071024                  JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
11572  070064  010200                          MOV     R2,R0
11573  070066  006200                          ASR     R0              ;DIVIDE BY TWO
11574  070070  103402                          BCS     0.OF1           ;BRANCH IF ODD
11575  070072  004537  071024                  JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
11576  070076  000137  067576          0.OF1:  JMP     0.DCD1          ;ALL DONE
11577                                   ;
11578                                   ; SEARCHES - $MSK   HAS THE MASK
11579                                   ;            $MSK+2 HAS THE FWA
11580                                   ;            $MSK+4 HAS THE LWA
11581
11582
11583
11584
11585                                   ;
11586  070102  005201                  0.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
11587  070104  000401                          BR      0.WDS
11588  070106  005001                  0.WSCH: CLR     R1              ;SET WORD SEARCH
11589  070110  005702                  0.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
11590  070112  001621                  0.ERR1: BEQ     0.ERR           ;ERROR IF NO OBJECT
11591  070114  013702  067270                  MOV     0.MSK+2,R2      ;SET ORIGIN
11592  070120  013705  067266                  MOV     0.MSK,R5        ;SET MASK
11593  070124  005105                          COM     R5              ;AND COMPLEMENT IT
11594  070126  020237  067272          0.WDS2: CMP     R2,0.MSK+4      ; IS THE SEARCH ALL DONE?
11595  070132  101217                          BHI     0.DCD           ;  YES
11596  070134  011200                          MOV     @R2,R0          ; GET OBJECT
11597  070136  005701                          TST     R1              ;NO
11598  070140  001027                          BNE     0.EFF1          ;BRANCH IF EFFECTIVE SEARCH
11599  070142  010046                          MOV     R0,-(SP)
11600  070144  010403                          MOV     R4,R3           ;EXCLUSIVE OR
11601  070146  040400                          BIC     R4,R0           ; IS DONE
11602  070150  042603                          BIC     (SP)+,R3                ;  IN A VERY
11603  070152  050003                          BIS     R0,R3           ;    FANCY MANNER HERE
11604  070154  040503                          BIC     R5,R3           ;AND RESULT WITH MASK
11605  070156  001016                  0.WDS3: BNE     0.WDS4          ;RE-LOOP IF NO MATCH
11606  070160  010446                          MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
11607  070162  004537  071260                  JSR     5,0.CRLF        ;TYPE <CR,LF>
11608  070166  010200                          MOV     R2,R0           ;GET READY TO TYPE
```

```
11609  070170  004537  071024                JSR     5,0.CADV        ;  TYPE ADDRESS
11610  070174  012700  000057                MOV     #'/,R0          ;SLASH TO R0
11611  070200  004537  071166                JSR     5,0.FTYP        ;TYPE IT
11612  070204  011200                        MOV     @R2,R0          ;GET CONTENTS
11613  070206  004537  071024                JSR     5,0.CADV        ;TYPE CONTENTS
11614  070212  012604                        MOV     (SP)+,R4        ; RESTORE R4
11615  070214  005722              0.WDS4:   TST     (R2)+           ;INCREMENT TO NEXT CELL AND
11616  070216  000743                        BR      0.WDS2          ;    RETURN
11617  070220  020004              0.EFF1:   CMP     R0,R4           ; IS (X)=K?
11618  070222  001755                        BEQ     0.WDS3          ;TYPE IF EQUAL
11619  070224  010003                        MOV     R0,R3           ;(X) TO R3
11620  070226  060203                        ADD     R2,R3           ;(X)+X
11621  070230  005203                        INC     R3
11622  070232  005203                        INC     R3              ;(X)+X+2
11623  070234  020304                        CMP     R3,R4           ;IS (X)+X+2=K?
11624  070236  001747                        BEQ     0.WDS3          ;BRANCH IF EQUAL
11625  070240  042700  177400                BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
11626  070244  110000                        MOVB    R0,R0           ;EXTEND SIGN
11627  070246  000257                        CCC
11628  070250  006300                        ASL     R0              ;MULTIPLY BY TWO
11629  070252  005200                        INC     R0              ;ADD TWO
11630  070254  005200                        INC     R0
11631  070256  060200                        ADD     R2,R0           ;ADD PC
11632  070260  020004                        CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
11633  070262  000735                        BR      0.WDS3
11634                                  ;
11635                                  ; PROCESS G - GO
11636                                  ;
11637  070264  105037  071307        0.GO:   CLRB    0.P             ;DISALLOW PROCEED
11638  070270  006204                        ASR     R4              ;CHECK LOW ORDER BIT
11639  070272  103617                        BCS     0.ERR2          ;ERROR IF ODD NUMBER
11640  070274  006304                        ASL     R4              ;RESTORE WORD
11641  070276  010437  067260                MOV     R4,0.UPC        ;SET UP NEW PC
11642  070302  112737  000340  177776        MOVB    #0.STM,ST       ;SET HIGH PRIORITY
11643  070310  004537  070756                JSR     5,0.RSTT        ;RESTORE TELETYPE
11644  070314  105037  071306        0.TBIT: CLRB    0.T             ;CLEAR BOTH
11645  070320  042737  000020  067262        BIC     #0.TBT,0.UST    ;   T-BIT FLAGS
11646  070326  017737  176742  067300        MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
11647  070334  013777  071350  176732        MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
11648  070342  012600              0.GO2:    MOV     (SP)+,R0        ;RESTORE
11649  070344  012601                        MOV     (SP)+,R1        ; R0
11650  070346  012602                        MOV     (SP)+,R2        ;    THRU
11651  070350  012603                        MOV     (SP)+,R3
11652  070352  012604                        MOV     (SP)+,R4        ;
11653  070354  012605                        MOV     (SP)+,R5        ;    R5
11654  070356  012606                        MOV     (SP)+,SP        ; AND SP
11655  070360  013746  067262                MOV     0.UST,-(SP)     ;    AND STATUS
11656  070364  013746  067260                MOV     0.UPC,-(SP)     ;    AND PC
11657  070370  000006              0.RTIT:   RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
11658                                  ;
11659                                  ; PROCESS P - PROCEED
11660                                  ;   ONLY ALLOWED AFTER A BREAKPOINT
11661                                  ;
11662  070372  105737  071307        0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
11663  070376  001645                        BEQ     0.ERR1          ;NOT LEGAL
11664  070400  105037  071307                CLRB    0.P             ;CLEAR PROCEED FLAG
```

```
11665  070404  005702                          TST     R2              ;WAS COUNT SPECIFIED?
11666  070406  001402                          BEQ     0.PR1           ;NO
11667  070410  010437  067276                  MOV     R4,0.CT         ;YES, PUT AWAY COUNT
11668  070414  112737  000340  177776  0.PR1:  MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
11669  070422  004537  070756                  JSR     5,0.RSTT        ;RESTORE TTY
11670  070426  112737  000340  177776  0.C1:   MOVB    #0.STM,ST       ;SET HIGH PRIORITY
11671  070434  105237  071306                  INCB    0.T             ;SET T-BIT FLAG
11672  070440  052737  000020  067262          BIS     #0.TBT,0.UST    ;SET T-BIT
11673  070446  000735                          BR      0.G02
11674                                  ;
11675                                  ; BREAKPOINT HANDLER
11676                                  ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
11677                                  ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
11678                                  ;  AND GIVES CONTROL TO THE COMMAND DECODER
11679                                  ;
11680  070450  012637  067260          0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
11681  070454  012637  067262                  MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
11682  070460  004037  070666          0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS'
11683  070464  105737  071306                  TSTB    0.T             ;CHECK FOR T-BIT SET
11684  070470  001311                          BNE     0.TBIT          ;JUMP IF SET
11685  070472  013777  067300  176574          MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
11686  070500  105737  067264                  TSTB    0.PRI           ;CHECK IF PRIORITY
11687  070504  100003                          BPL     0.BK2           ; IS AS SAME AS USER PGM
11688  070506  113705  067262                  MOVB    0.UST,R5        ;PICK UP USER UST IF SO
11689  070512  000407                          BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
11690  070514  113705  067264          0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
11691  070520  000257                          CCC                     ;CLEAR CARRY
11692  070522  106005                          RORB    R5              ;SHIFT LOW ORDER BITS
11693  070524  106005                          RORB    R5              ;   INTO
11694  070526  106005                          RORB    R5              ;     HIGH ORDER
11695  070530  106005                          RORB    R5              ;       POSITION
11696  070532  110537  177776          0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
11697  070536  013705  067260                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
11698  070542  005745                          TST     -(R5)           ;SUBTRACT TWO
11699  070544  010537  067260                  MOV     R5,0.UPC        ;FROM THE USER'S PC
11700  070550  020537  067274                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
11701  070554  001417                          BEQ     0.B2            ;JUMP IF FOUND
11702  070556  004537  070724                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
11703  070562  004537  071260                  JSR     5,0.CRLF
11704  070566  012704  071312                  MOV     #0.BD,R4        ;ERROP, NOTHING FOUND
11705  070572  012703  071313                  MOV     #0.BD+1,R3
11706  070576  004537  071152                  JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
11707  070602  010500                          MOV     R5,R0
11708  070604  042737  000020  067262          BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
11709  070612  000420                          BR      0.B3            ; AND CONTINUE
11710  070614  005337  067276          0.B2:   DEC     0.CT
11711  070620  003302                          BGT     0.C1            ;JUMP IF REPEAT
11712  070622  012737  000001  067276          MOV     #1,0.CT         ;RESET COUNT TO 1
11713  070630  105237  071307                  INCB    0.P             ;ALLOW PROCEED
11714  070634  004537  070724                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
11715  070640  012700  000102                  MOV     #'B,R0
11716  070644  004537  071166                  JSR     5,0.FTYP        ;TYPE "B"
11717  070650  013700  067274                  MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
11718  070654  004537  071024          0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
11719  070660  005005                          CLR     R5              ;CLEAR CAD
11720  070662  000137  067572                  JMP     0.DCD           ;GO TO DECODER
```

```
11721                                   ;
11722                                   ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
11723                                   ;
11724  070666  012637  071304          0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
11725  070672  010637  067256                  MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
11726  070676  012706  067256                  MOV     #0.USP,SP       ;SET TO INTERNAL STACK
11727  070702  010546                          MOV     R5,-(SP)        ;SAVE
11728  070704  010446                          MOV     R4,-(SP)        ; REGISTERS
11729  070706  010346                          MOV     R3,-(SP)        ;1
11730  070710  010246                          MOV     R2,-(SP)        ; THRU
11731  070712  010146                          MOV     R1,-(SP)        ;     5
11732  070714  013746  071304                  MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
11733  070720  005746                          TST     -(SP)
11734  070722  000200                          RTS     R0
11735                                   ;
11736                                   ; SAVE TELETYPE STATUS
11737                                   ;
11738  070724  113737  177560  071310  0.SVTT: MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
11739  070732  113737  177564  071311          MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
11740  070740  105037  177560                  CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
11741  070744  105037  177564                  CLRB    0.TCSR          ; BITS IN BOTH C/SR
11742  070750  004537  071260                  JSR     5,0.CRLF        ;TYPE <CR,LF>
11743  070754  000205                          RTS     R5
11744                                   ;
11745                                   ; RESTORE TELETYPE STATUS
11746                                   ;
11747  070756  004537  071260          0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
11748  070762  105737  177564                  TSTB    0.TCSR          ;WAIT READY ON PRINTER
11749  070766  100375                          BPL     .-4
11750  070770  032737  004000  177560          BIT     #4000,0.RCSR    ;CHECK BUSY FLAG ON READER
11751  070776  001403                          BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
11752  071000  105737  177560                  TSTB    0.RCSR          ;WAIT READY
11753  071004  100375                          BPL     .-4             ;     ON READER
11754  071006  113737  071310  177560  0.RSE1: MOVB    0.CSR1,0.RCSR   ;RESTORE
11755  071014  113737  071311  177564          MOVB    0.CSR2,0.TCSR   ;  THE STATUS REGISTERS
11756  071022  000205                          RTS     R5
11757                                   ;
11758                                   ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
11759                                   ; WORD IS IN R0
11760                                   ;
11761  071024  010246                  0.CADV: MOV     R2,-(SP)        ;SAVE R2
11762  071026  012704  071347                  MOV     #0.BUF+6,R4     ;BUFFER START ADDRESS
11763  071032  012746  000060                  MOV     #'0,-(SP)               ;CONSTANT ASCII 0
11764  071036  010002                  0.SPC:  MOV     R0,R2           ; GET
11765  071040  042702  177770                  BIC     #177770,R2      ;     OCTAL CHARACTER
11766  071044  061602                          ADD     @SP,R2          ;CONVERT TO ASCII
11767  071046  110244                          MOVB    R2,-(R4)        ;STORE IN BUFFER
11768  071050  006200                          ASR     R0              ;SHIFT THIS MESS
11769  071052  006200                          ASR     R0              ; RIGHT
11770  071054  006200                          ASR     R0              ;   THREE WHOLE PLACES
11771  071056  020427  071342                  CMP     R4,#0.BUF+1     ;DONE?
11772  071062  101365                          BHI     0.SPC           ;     NO
11773  071064  042700  177776                  BIC     #177776,R0      ;GET LAST BIT
11774  071070  062600                          ADD     (SP)+,R0                ;CONVERT TO ASCII
11775  071072  110044                          MOVB    R0,-(R4)        ;AND PUT IT AWAY
11776  071074  012703  071347                  MOV     #0.BUF+6,R3     ;LWA
```

```
11777  071100  004537  071152                JSR     5,0.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
11778  071104  012602                        MOV     (SP)+,R2        ;RESTORE R2
11779  071106  000205                        RTS     R5
11780                                  ;
11781                                  ; GENERAL CHARACTER INPUT ROUTINE
11782                                  ; CHARACTER INPUT GOES TO R0
11783                                  ;
11784  071110  105737  177560          0.GET:  TSTB    0.RCSR          ;WAIT FOR
11785  071114  100375                        BPL     .-4             ;    INPUT FROM KEYBOARD
11786  071116  113700  177562                MOVB    0.RDB,R0        ;GET A CHARACTER
11787  071122  004537  071166                JSR     5,0.FTYP        ;ECHO CHARACTER
11788  071126  042700  177600                BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
11789  071132  001766                        BEQ     0.GET           ;IGNORE NULLS
11790  071134  122700  000040                CMPB    #40,R0          ;CHECK FOR SPACES
11791  071140  001763                        BEQ     0.GET           ;IGNORE NULLS
11792  071142  122700  000073                CMPB    #';,R0          ;CHECK FOR SEMI-COLON
11793  071146  001760                        BEQ     0.GET           ;IGNORE THEM IF FOUND
11794  071150  000205                        RTS     R5
11795                                  ;
11796                                  ; GENERAL CHARACTER OUTPUT ROUTINE
11797                                  ;  ADDRESS OF FIRST BYTE IN R4,
11798                                  ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
11799                                  ;
11800  071152  020304                  0.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
11801  071154  103426                        BLO     0.TYP1          ;  EXIT WHEN DONE
11802  071156  112400                        MOVB    (R4)+,R0        ;GET A CHARACTER
11803  071160  004537  071166                JSR     5,0.FTYP        ;TYPE ONE CHARACTER
11804  071164  000772                        BR      0.TYPE          ;LOOP UNTIL DONE
11805                                  ;
11806                                  ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
11807                                  ;
11808  071166  105737  177564          0.FTYP: TSTB    0.TCSR          ;CHECK STATUS
11809  071172  100375                        BPL     .-4             ;WAIT UNTIL READY
11810  071174  110037  177566                MOVB    R0,0.TDB        ;TYPE ONE CHARACTER
11811  071200  120037  000045                CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
11812  071204  001012                        BNE     0.TYP1          ;NO
11813  071206  113746  000044                MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
11814  071212  105737  177564          0.TYP2: TSTB    0.TCSR
11815  071216  100375                        BPL     0.TYP2
11816  071220  105037  177566                CLRB    0.TDB           ;GENEPATE NULL FILLER
11817  071224  105316                        DECB    @SP
11818  071226  003371                        BGT     0.TYP2
11819  071230  005726                        TST     (SP)+           ;POP STACK
11820  071232  000205                  0.TYP1: RTS     R5
11821                                  ;
11822                                  ; CLOSE WORD OR BYTE AND EXIT,
11823                                  ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
11824                                  ;
11825  071234  006205                  0.TCLS: ASR     R5              ;GET LOW ORDER BIT
11826  071236  103405                        BCS     0.TC            ;JUMP IF ALREADY CLOSED
11827  071240  006305                        ASL     R5
11828  071242  005702                        TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
11829  071244  001401                        BEQ     0.CLS1          ;NO CHANGE TO THE OPEN CELL
11830  071246  010415                        MOV     R4,@R5          ;STORE WORD
11831  071250  000207                  0.CLS1: RTS     PC
11832  071252  005746                  0.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
```

```
11833  071254  000137  067556                  JMP     0.ERR           ;AND SCREAM BLOODY MURDER
11834                                  ;
11835                                  ; 0.CRLF - TYPE <CR,LF>
11836                                  ; 0.CRLS - TYPE <CR,LF>*
11837                                  ;
11838  071260  012703  071315          0.CRLF: MOV     #0.CR+1,R3      ;LWA <CR,LF>
11839  071264  000402                          BR      0.CRS
11840  071266  012703  071316          0.CRLS: MOV     #0.CR+2,R3      ;LWA <CR,LF>*
11841  071272  012704  071314          0.CRS:  MOV     #0.CR,R4        ;FWA
11842  071276  004537  071152                  JSR     5,0.TYPE        ;TYPE SOMETHING
11843  071302  000205                          RTS     R5
11844                                  ;
11845  071304  000000                  0.XXX:  .WORD   0               ;TEMPORARY STORAGE
11846  071306     000                  0.T:    .BYTE   0               ;  T-BIT FLAG
11847  071307     000                  0.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
11848                                  ;                               = 1 IF PROCEED ALLOWED
11849  071310     000                  0.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
11850  071311     000                  0.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
11851                                  ;
11852                                          .EVEN
11853  071312  042502                  0.BD:   .WORD   "BE
11854                                  ;
11855  071314     015                  0.CR:   .BYTE   015     ;  <CR>
11856  071315     012                          .BYTE   012     ;  <LF>
11857  071316     052                          .BYTE   '*      ;  *
11858                                  ;
11859  071317     057                  0.LGCH: .BYTE   '/      ;  /
11860  071320     015                          .BYTE   015     ;  CARRIAGE RETURN
11861  071321     044                          .BYTE   '$      ;  $
11862  071322     107                          .BYTE   'G      ;  G
11863  071323     012                          .BYTE   012     ;  <LF>
11864  071324     137                          .BYTE   '-      ;  -
11865  071325     136                          .BYTE   '       ;
11866  071326     117                          .BYTE   'O      ;  O
11867  071327     127                          .BYTE   'W      ;  W
11868  071330     105                          .BYTE   'E      ;  E
11869  071331     102                          .BYTE   'B      ;  B
11870  071332     120                          .BYTE   'P      ;  P
11871          000014          0.CLGT  =       .-0.LGCH                ,TABLE LENGTH
11872                                  ;
11873  071333     123          0.TL:   .BYTE   'S      ;DO        1
11874  071334     120                  .BYTE   'P      ;NOT       2
11875  071335     115                  .BYTE   'M      ;CHANGE    3
11876  071336     000                  .BYTE   0       ;THE       4
11877  071337     000                  .BYTE   0       ;ORDER     5
11878  071340     102                  .BYTE   'B      ;HERE      6
11879          000006          0.LG    =       .-0.TL
11880                                  ;
11881  071341                  0.BUF:                          ;6 CHAR. BUFFER WITH
11882          071347                  =       .+6
11883  071347     040                  .BYTE   '               ;TRAILING BLANK
11884                                          .EVEN
11885                                  ;
11886  071350  000003          0.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
11887                                  ;
11888                                  ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
```

```
11889                                   ;
11890           067242                  .       =       0.ODT-40
11891   067242  000000                  0.URO:  0       ;USER R0
11892   067244  000000                          0       ;     R1
11893   067246  000000                          0       ;     R2
11894   067250  000000                          0       ;     R3
11895   067252  000000                          0       ;     R4
11896   067254  000000                          0       ;     R5
11897   067256  000000                  0.USP:  0       ;USER SP
11898   067260  000000                  0.UPC:  0       ;USER PC
11899   067262  000000                  0.UST:  0       ;USER ST
11900   067264  000007                  0.PRI:  7       ;ODT PRIORITY
11901   067266  000000                  0.MSK:  0       ;MASK
11902   067270  000000                          0       ;LOW LIMIT
11903   067272  000000                          0       ;HIGH LIMIT
11904                                   ;
11905                                   ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
11906                                   ;   UIN = CONTENTS
11907                                   ;
11908   067274  000000                  0.ADR1: 0
11909   067276  000000                  0.CT:   0
11910   067300  000000                  0.UIN:  0
11911           000001                          .END
```

```
ABASE = 177440           2553    2594    2608#
ACDW1 = 000000           2553    2596
ACDW2 = 000000           2553    2597
ACL0  = 000010           1434#
ACPUOP= 000000           2553    2568
ACT11   005430           2763#   3734*
ADDW0 = 000000           2553    2598
ADDW1 = 000000           2553    2599
ADDW10= 000000           2553
ADDW11= 000000           2553
ADDW12= 000000           2553
ADDW13= 000000           2553
ADDW14= 000000           2553
ADDW15= 000000           2553
ADDW2 = 000000           2553    2600
ADDW3 = 000000           2553    2601
ADDW4 = 000000           2553    2602
ADDW5 = 000000           2553    2603
ADDW6 = 000000           2553    2604
ADDW7 = 000000           2553    2605
ADDW8 = 000000           2553
ADDW9 = 000000           2553
ADEVCT= 000000           2553    2559
ADEVM = 000000           2553    2595
AENV  = 000000           2553    2564
AENVM = 000000           2553    2565
AFATAL= 000000           2553    2556
AMADR1= 000000           2553    2581
AMADR2= 000000           2553    2585
AMADR3= 000000           2553    2588
AMADR4= 000000           2553    2591
AMAMS1= 000000           2553    2575
AMAMS2= 000000           2553    2583
AMAMS3= 000000           2553    2586
AMAMS4= 000000           2553    2589
AMSGAD= 000000           2553    2561
AMSGLG= 000000           2553    2562
AMSGTY= 000000           2553    2555
AMTYP1= 000000           2553    2576
AMTYP2= 000000           2553    2584
AMTYP3= 000000           2553    2587
AMTYP4= 000000           2553    2590
APASS = 000000           2553    2558
APRIOR= 000000           2553
APTCSU= 000040           9362    9534#
APTENV= 000001           9309    9355    9490    9532#
APTSIZ= 000200           3663    9531#
APTSPO= 000100           9357    9492    9533#
ASWREG= 000000           2553    2566
ATESTN= 000000           2553    2557
ATTN    005304           2698#   7141    7269    7732    7854    8261    8281    8308    8337
AUNIT = 000000           2553    2560
AUSWR = 000000           2553    2567
AVECT1= 000000           2553    2592
AVECT2= 000000           2553    2593
BADHDR  005300           2688#   3742*   9003
```

```
BADTMO  047736          3817    9072#
BAI   = 000020          1397#   7483    7605
BA16  = 000400          1383#
BA17  = 001000          1384#
BIT0  = 000001          1326#   1380    1412    1431    2755    3858    3860    8117    8944
BIT00 = 000001          1316#   1326
BIT01 = 000002          1315#   1325
BIT02 = 000004          1314#   1324
BIT03 = 000010          1313#   1323
BIT04 = 000020          1312#   1322
BIT05 = 000040          1311#   1321
BIT06 = 000100          1310#   1320
BIT07 = 000200          1309#   1319
BIT08 = 000400          1308#   1318    9244
BIT09 = 001000          1307#   1317    9252    9320
BIT1  = 000002          1325#   1413    2756
BIT10 = 002000          1306#   1385    1403    1422    1454    1468    1482    1495    1509    9297
BIT11 = 004000          1305#   1386    1404    1423    1440    1455    1469    1483    1496    1510    9259
BIT12 = 010000          1304#   1387    1405    1424    1456    1470    1484    1497    1511
BIT13 = 020000          1303#   1388    1406    1425    1441    1457    1471    1485    1498    1512    9304
BIT14 = 040000          1302#   1389    1407    1426    1442    1458    1472    1486    1499    1513    1523    8755    9230
BIT15 = 100000          1301#   1390    1391    1408    1427    1443    1459    1526    8751
BIT2  = 000004          1324#   1414    1433    2757
BIT3  = 000010          1323#   1396    1415    1434
BIT4  = 000020          1322#   1397    1416    1435    1448    1476    1503
BIT5  = 000040          1321#   1398    1417    1436    1449    1463    1477    1490    1504
BIT6  = 000100          1320#   1381    1399    1418    1437    1450    1464    1478    1491    1505
BIT7  = 000200          1319#   1382    1400    1419    1438    1451    1465    1479    1492    1506    3848
BIT8  = 000400          1318#   1383    1401    1420    1439    1452    1466    1480    1493    1507
BIT9  = 001000          1317#   1384    1402    1421    1453    1467    1481    1494    1508
BPTVEC= 000014          1333#
BSE   = 000200          1419#   7493
BSERR   001452          2673#   7405*   7410*
BSE22H  002276          2682#   7355    7407    7426    8749    8847
BSE22S  003276          2683#   7417    8753    8851
BYP     044260          3940    3968    3972    3987    4048    4075    4079    4083    4087    8359#
BYPCER  001456          2675#   3845*   4100*   8478
CALADD  001362          2636#   8721    8727    8729    8776
CALDIF  001354          2633#   4745*   4840*  11063   11087
CCLR  = 100000          1391#   4232    4259    4299    4381    4396    4433    4540    4555    4592    4754    4849    5012
                        5187    5516    5617    5741    5842    5941    6074    6710    6738    6793    6821    6903    6995
                        7110    7150    7238    7278    7503    7722    7741    7843    7863    8519    8528    8537    8546
                        9007    9023
CCYL    001350          2631#
CDT   = 002000          1385#   3956    4058    4142    4144
CERR  = 100000          1390#   3919    4032    4360    4519    4719    4814    4916    4968    5091    5143    5480    5508
                        5705    5733    5933    6066    7088    7216    7368    7490    7616    7754    7784    7875    7905
                        7958    8480    8581
CFMT  = 010000          1387#   8843
CHKFLG  001460          2676#   8369*   8456    8459    8463
CHKMSG  044274          4283    4318    4347    4370    4416    4452    4506    4529    4575    4611    4733    4773    4828
                        4868    4930    4953    4982    5005    5031    5055    5105    5128    5157    5180    5206    5230
                        5296    5317    5378    5399    5463    5494    5550    5589    5636    5688    5719    5775    5814
                        5861    5918    6000    6051    6133    6201    6258    6342    6383    6462    6503    6566    6638
                        7102    7129    7230    7257    7380    7515    7534    7635    7654    7768    7807    7889    7928
                        7972    8369#   9042
```

```
CKCERR  045064          8210  8214  8225  8230  8478#
CKSWR = 104407          9229  9293  9319  10194#
CLEAR = 000005          1368#  4302  4436  4595  4757  4852  5015  5190  5620  5845  7113  7241  9026
CLKOF   047162          4402  4474  4561  4633  6584  6656  6746  6829  8904#  8919
CLKON   047114          4385  4544  6573  6645  6740  6823  8888#  8915
CLOCK   047154          3766  8899#
CLRFLG  043026          3684  8062#
COE   = 001000          1421#
COUNT   001366          2640#  4222*  4248*  4328*  4384*  4461  4487*  4543*  4620  4700*  4749*  4795*  4844*
                        4898*  4937*  4989*  5038*  5073*  5112*  5164*  5213*  5253*  5335*  5643*  5868*  5984*
                        6117*  6162*  6219*  6553*  6572*  6585*  6586  6625*  6644*  6657*  6658  6739*  6747*
                        6748  6822*  6830*  6831  6894*  6986*  7161*  7289*  8339  8899*  8916
CR    = 000015          1241#  9401  9411  10205  10213  10218  10222  10232  10241  10248  10256  10263  10268
                        10272  10278  10282  10288  10294  10299  10308  10316  10323  10331  10336  10343  10348
                        10353  10358  10362  10368  10373  10377  10381  10383  10387  10392  10394  10395  10399
                        10406  10410  10417  10422  10426  10432  10450  10458  10478  10486  10549  10576  10583
                        10588  10593  10605  10626  10642  10810  10818  10847  10880  10998
CRLF  = 000200          1242#  9372  9411
CTO   = 004000          1386#  8484
CYLADD  001360          2635#  4294  4741  4836  7478*  7482  7598*  7602*  7607  8620*  8621*  8622*  8623*
                        8624*  8625*  8643*  8644*  8645*  8646*  8647*  8648*
CYLDIF  001356          2634#  4291  8604*  8605*  8606*  8607*  8608*  8609*  8610  8612*  8630*  8631*  8632*
                        8633*  8634*  8635*  8636  8638*
DATAO   001434          2664#  7479  7596  7603  7621
DATAO1  001436          2665#  7479*  7481  7596  7599*  7603*  7606
DATA1   001440          2666#  7599
DATCMD  043430          7365  7487  7613  7782  7903  8190#
DCK   = 100000          1427#
DCLO  = 000020          1435#
DCPAR = 020000          1388#
DDISP = 177570          1248#  2528  3651
DDPCH   005426          2762#  3730*  3927  4135
DDT   = 000400          1439#
DDUMP   005424          2761#  3697*  8062
DF1     066332          2806  2812  2818  2824  2835  2841  2846  3074  3114  3119  11139#
DF10    066462          2851  2856  2861  2866  2871  2893  2898  2908  2924  2984  2994  2999  3034
                        3089  3094  3099  3104  3109  3144  3169  3174  3179  3184  3189  3194  3199
                        3204  3209  3214  3219  3224  3229  3249  3254  3259  3284  3309  3334  3339
                        3344  3349  3354  3359  3364  3369  3414  3419  3434  3439  3444  3464  3489
                        3544  3559  3594  11190#
DF14    066502          3459  11199#
DF15    066522          3424  3429  3449  3454  3564  11209#
DF17    066542          3569  3584  11218#
DF2     066346          2881  11146#
DF20    066566          3014  11228#
DF21    066612          2903  2913  2918  2929  2934  2939  2944  2949  2954  2959  2964  2969  2974
                        2979  3019  3024  3029  3039  3044  3049  3054  3059  3064  3069  3079  3084
                        3124  3129  3134  3139  3149  3154  3159  3164  3264  3269  3274  3279  3289
                        3294  3299  3304  3314  3319  3324  3329  3374  3379  3384  3389  3394  3399
                        3404  3409  3494  3499  3504  3509  3534  3539  11239#
DF22    066646          3004  3009  11254#
DF23    066702          11269#
DF3     066352          2876  2887  11149#
DF4     066376          2989  11160#
DF5     066412          3234  3239  3244  11167#
DF6     066436          3479  3484  3549  11178#
```

```
DH1     062556          2804    2810    2816    2822    2833    2839    2844    2987    3072    3112    3117    3457    10775#
                        11151   11171   11180   11192   11211   11222   11230   11241   11256   11271
DH10    063310          3107    3432    3437    10837#
DH11    063337          10841#  11153
DH12    063354          3547    10844#
DH13    063445          3567    10855#
DH14    063531          3142    10864#
DH15    063563          3147    3152    3157    3162    10869#
DH16    063612          3167    10873#
DH17    063741          3002    3007    3032    3077    3082    3227    3532    3537    10888#
DH18    063765          3102    10892#
DH19    064012          2922    3197    10896#
DH2     062573          10778#  11141   11155   11162   11173   11184   11194   11203   11213   11224   11234   11249   11264
                        11279
DH20    064035          3202    10900#
DH21    064070          2906    3207    10905#
DH22    064116          2942    2947    2952    2957    3182    3332    3347    10909#
DH23    064150          3172    10914#
DH24    064200          3187    10918#
DH25    064245          3012    3252    3257    3282    3357    3362    3372    3377    3382    3387    3477    3482    3487
                        10925#
DH26    064270          2859    2864    2884    2890    2916    2927    3027    3037    3557    3562    10929#
DH27    064320          2849    2854    2901    2911    2982    2997    3017    3022    10933#
DH28    064351          10938#  11243   11258   11273
DH29    064361          10940#  11247   11262   11277
DH3     062643          10785#  11143   11157   11164   11175   11186   11196   11205   11215   11226   11236   11251   11266
                        11281
DH30    064371          3052    3057    3062    3067    3412    3422    3427    10942#
DH31    064423          10947#  11245   11260   11275
DH32    064450          2869    2874    2896    2932    2937    3042    3047    10951#
DH35    064502          3092    3192    3222    3262    3267    3272    3277    3417    3492    3497    3502    3507    10956#
DH36    064527          3247    10960#
DH37    064610          3217    10969#
DH38    064664          3312    3317    3322    3327    3337    3367    10977#
DH39    064736          2962    2967    2972    2977    3447    3452    10984#
DH4     062714          3097    10792#
DH40    064771          10989#  11201
DH41    065025          3087    3442    3462    10994#
DH42    065121          3582    11005#
DH43    065172          3342    11012#
DH44    065251          3592    11020#  11220
DH45    065330          3352    11028#
DH49    065374          11034#  11169
DH5     062760          2992    3122    3127    3132    3137    3177    3307    3392    3397    3402    3407    10798#
DH51    065451          3287    3292    3297    3302    11042#
DH56    065504          11047#  11232
DH6     063007          10802#  11182
DH7     063042          3542    10807#
DH8     063214          2879    10826#
DH9     063235          3212    10829#
DI    = 040000          1389#
DISPLA  001142          2528#   3651*   3659*   9274*   9296*
DISPRE  000174          1535#   3659
DLT   = 100000          1408#   7793    7914
DLY     044242          3915    4028    8350#   8353
DMD   = 000040          1449#
```

```
DOCMD   043372          4236  4263  4303  4437  4596  4712  4758  4807  4853  4909  4961  5016  5084
                        5136  5191  5268  5276  5350  5358  5439  5447  5473  5523  5531  5562  5570
                        5601  5609  5621  5664  5672  5698  5748  5756  5787  5795  5826  5834  5846
                        5894  5902  5948  5956  5969  5977  6027  6035  6081  6089  6102  6110  6177
                        6185  6234  6242  6286  6294  6306  6314  6322  6354  6362  6406  6414  6426
                        6434  6442  6474  6482  6535  6543  6607  6615  6686  6694  6717  6725  6769
                        6777  6800  6808  6867  6875  6910  6918  6931  6959  6967  7002  7010  7023
                        7059  7067  7081  7114  7187  7195  7209  7242  7332  7340  7459  7467  7550
                        7558  7575  7583  7703  7711  7727  7828  7836  7848  7951  7987  7995  8179#
                        8509  8993  9011  9027
DPAT1   001442          2667#
DPAT2   001444          2668#
DRA   = 000001          1431#  3962  4065
DRAV    044172          4229  4254  4335  4356  4387  4392  4403  4494  4515  4546  4551  4562  4707
                        4780  4802  4875  4905  4941  4993  5080  5116  5168  5260  5342  5504  5729
                        5929  6062  6169  6226  6555  6575  6627  6647  6706  6734  6789  6817  6898
                        6990  7146  7153  7165  7274  7281  7293  7736  7751  7858  7873  8326#
DRDY  = 000200          1438#
DRIVS   005434          2765#  3853*  3866*  3883  3899*  3929*  3948  4052*  4119  8025  8120*
DRIVO   005436          2771#  3746  3855  3901  4014  8101
DRIV1   005440          2772#
DRIV2   005442          2773#
DRIV3   005444          2774#
DRIV4   005446          2775#
DRIV5   005450          2776#
DRIV6   005452          2777#
DRIV7   005454          2778#
DROT  = 000040          1436#
DRPAR = 000010          1415#
DRVMSK= 000007          1395#  3922  4035
DRVPTR  001342          2626#  3746*  4126  4146*
DSC   = 040000          1442#
DSWR  = 177570          1247#  2527  3650
DTE   = 010000          1424#
DTYE  = 000040          1417#  3954  4056
DT1     065530          2805  2811  2817  2823  2834  2840  2845  2850  2855  2860  2865  2870  2891
                        2897  2907  2923  2983  2988  2993  2998  3033  3073  3088  3093  3098  3103
                        3108  3113  3118  3143  3168  3173  3178  3183  3188  3193  3198  3203  3208
                        3213  3218  3223  3228  3233  3238  3243  3248  3253  3258  3283  3308  3333
                        3338  3343  3348  3353  3358  3363  3368  3413  3418  3423  3428  3433  3438
                        3443  3448  3453  3463  3488  3513  3558  3563  3568  3583  3593  11055#
DT13    066066          2902  2912  2917  2928  2933  2938  2943  2948  2953  2958  2963  2968  2973
                        2978  3018  3023  3028  3038  3043  3048  3053  3058  3063  3068  3078  3083
                        3123  3128  3133  3138  3148  3153  3158  3163  3263  3268  3273  3278  3288
                        3293  3298  3303  3313  3318  3323  3328  3373  3378  3383  3388  3393  3398
                        3403  3408  3493  3498  3503  3508  3533  3538  11103#
DT14    066146          3003  3008  11113#
DT15    066236          11124#
DT3     065570          2880  11062#
DT4     065574          3483  3548  11063#
DT6     065642          2875  2885  11071#
DT7     065706          3458  11079#
DT8     065754          3478  11087#
DT9     066022          3013  11095#
D.ACLO= 000100          1491#
D.BRHM= 000100          1478#  4277  4312  4341  4410  4446  4500  4569  4605  4727  4767  4822  4862
```

```
                    4924    4947    4976    4999    5025    5049    5099    5122    5151    5174    5200    5224    5290
                    5311    5372    5393    5457    5488    5544    5583    5630    5682    5713    5769    5808    5855
                    5912    5994    6045    6127    6195    6252    6336    6377    6456    6497    6560    6632    7096
                    7123    7224    7251    7374    7509    7528    7629    7648    7762    7801    7883    7922    7966
                    9036
D. CART= 000400     1480#   4277    4312    4341    4410    4446    4500    4569    4605    4727    4767    4822    4862
                    4924    4947    4976    4999    5025    5049    5099    5122    5151    5174    5200    5224    5290
                    5311    5372    5393    5457    5488    5544    5583    5630    5682    5713    5769    5808    5855
                    5912    5994    6045    6127    6195    6252    6336    6377    6456    6497    6560    6632    7096
                    7123    7224    7251    7374    7509    7528    7629    7648    7762    7801    7883    7922    7966
                    9036
D. DDT = 000400     1466#   4167
D. DOOR= 000200     1479#   4277    4312    4341    4410    4446    4500    4569    4605    4727    4767    4822    4862
                    4924    4947    4976    4999    5025    5049    5099    5122    5151    5174    5200    5224    5290
                    5311    5372    5393    5457    5488    5544    5583    5630    5682    5713    5769    5808    5855
                    5912    5994    6045    6127    6195    6252    6336    6377    6456    6497    6560    6632    7096
                    7123    7224    7251    7374    7509    7528    7629    7648    7762    7801    7883    7922    7966
                    9036
D. DRA = 000040     1463#   4275    4310    4339    4408    4444    4498    4567    4603    4725    4765    4820    4860
                    4922    4945    4974    4997    5023    5047    5097    5120    5149    5172    5198    5222    5279
                    5288    5309    5361    5370    5391    5450    5455    5486    5534    5542    5573    5581    5612
                    5628    5675    5680    5711    5759    5767    5798    5806    5837    5853    5905    5910    5959
                    5980    5992    6038    6043    6092    6113    6125    6188    6193    6245    6250    6297    6325
                    6334    6365    6375    6417    6445    6454    6485    6495    6546    6558    6618    6630    6697
                    6728    6780    6811    6878    6921    6970    7013    7070    7094    7121    7198    7222    7249
                    7343    7372    7470    7507    7526    7561    7586    7627    7646    7714    7760    7799    7839
                    7881    7920    7964    7998    8327    9034
D. DRDY= 000200     1465#   4275    4310    4339    4408    4444    4498    4567    4603    4725    4765    4820    4860
                    4922    4945    4974    4997    5023    5047    5097    5120    5149    5172    5198    5222    5288
                    5309    5370    5391    5455    5486    5542    5581    5628    5680    5711    5767    5806    5853
                    5910    5992    6043    6125    6193    6250    6334    6375    6454    6495    6558    6630    7094
                    7121    7222    7249    7372    7507    7526    7627    7646    7760    7799    7881    7920    7964
                    9034
D. DROT= 020000     1498#
D. DSC = 040000     1472#   4275    4408    4567    4725    4820    4922    4974    4997    5047    5097    5149    5172
                    5222    5486    5542    5581    5711    5767    5806    6334    6454    7094    7222    7760    7881
                    7964
D. FLT = 000200     1492#
D. FORM= 001000     1467#
D. FWD = 002000     1482#
D. HDFL= 000200     1506#
D. HDHM= 000040     1477#   8677
D. IDAE= 000040     1490#
D. ILF = 000400     1493#
D. LIMD= 020000     1512#
D. LOAD= 010000     1484#   8694
D. MHD = 000400     1507#
D. NMOV= 010000     1511#
D. OFF = 002000     1468#
D. PAR = 001000     1494#
D. PIP = 020000     1471#   7158    7286
D. PLO = 004000     1510#
D. REV = 004000     1483#
D. RTZ = 020000     1485#   4268    9016
D. SECT= 000020     1503#
D. SKI = 002000     1495#
```

```
D. SPIN= 010000   1470#  4275  4310  4339  4363  4408  4444  4498  4522  4567  4603  4725  4765
                  4820  4860  4922  4945  4974  4997  5023  5047  5097  5120  5149  5172  5198
                  5222  5288  5309  5370  5391  5455  5486  5542  5581  5628  5680  5711  5767
                  5806  5853  5910  5992  6043  6125  6193  6250  6334  6375  6454  6495  6558
                  6630  7094  7121  7222  7249  7372  7507  7526  7627  7646  7760  7799  7881
                  7920  7964  9034
D. SPLS= 010000   1497#
D. SPOK= 001000   1481#  4277  4312  4341  4410  4446  4500  4569  4605  4727  4767  4822  4862
                  4924  4947  4976  4999  5025  5049  5099  5122  5151  5174  5200  5224  5290
                  5311  5372  5393  5457  5488  5544  5583  5630  5682  5713  5769  5808  5855
                  5912  5994  6045  6127  6195  6252  6336  6377  6456  6497  6560  6632  7096
                  7123  7224  7251  7374  7509  7528  7629  7648  7762  7801  7883  7922  7966
                  9036
D. SSP  = 000020   1476#  4277  4312  4341  4410  4446  4500  4569  4605  4727  4767  4822  4862
                  4924  4947  4976  4999  5025  5049  5099  5122  5151  5174  5200  5224  5290
                  5311  5372  5393  5457  5488  5544  5583  5630  5682  5713  5769  5808  5855
                  5912  5994  6045  6127  6195  6252  6336  6377  6456  6497  6560  6632  7096
                  7123  7224  7251  7374  7509  7528  7629  7648  7762  7801  7883  7922  7966
                  9036
D. SUNS= 040000   1513#
D. TIB  = 002000   1509#
D. UNLD= 040000   1486#
D. UNS  = 040000   1499#
D. VV   = 000100   1464#  4239  4275  4310  4339  4363  4408  4444  4498  4522  4567  4603  4725
                  4765  4820  4860  4922  4945  4974  4997  5023  5047  5097  5120  5149  5172
                  5198  5222  5288  5309  5370  5391  5455  5486  5542  5581  5628  5680  5711
                  5767  5806  5853  5910  5992  6043  6125  6193  6250  6334  6375  6454  6495
                  6558  6630  7094  7121  7222  7249  7372  7507  7526  7627  7646  7760  7799
                  7881  7920  7964  9034
D. WCUR= 000040   1504#
D. WGAT= 000100   1505#
D. WLE  = 004000   1496#
D. WRL  = 004000   1469#
D. XERR= 001000   1508#
ECCW   = 020000   1457#
ECH    = 000100   1418#
EMTVEC= 000030   1336#  3634*  3635*
EM1     057054   8129  10432#
EM10    057730   2838  10510#
EM11    060013   2843  10519#
EM12    060057   2848  2858  2868  3196  3201  3206  3211  3226  3251  3331  3411  3446  3556
                 10526#
EM13    060115   3031  3101  3106  3166  3176  3216  3256  3336  3356  10532#
EM14    060137   2878  10536#
EM15    060175   2986  10542#
EM16    060304   3566  3581  10554#
EM17    060347   2941  2961  3016  3026  3041  3051  3121  3146  3261  3286  3311  3371  3391
                 3491  3531  10560#
EM18    060370   2900  2915  2931  2946  2966  3056  3076  3126  3151  3266  3291  3316  3376
                 3396  3496  10563#
EM19    060411   2951  2971  3021  3036  3046  3061  3131  3156  3271  3296  3321  3381  3401
                 3501  3536  10566#
EM2     057127   2803  10440#
EM20    060432   2910  2926  2936  2956  2976  3066  3081  3136  3161  3276  3301  3326  3386
                 3406  3506  10569#
EM21    060453   2853  2863  2895  2905  3426  3451  3486  3541  3561  10572#
```

```
EM22    060475          3111   10576#
EM23    060602          3116   10588#
EM24    060702          2921   10600#
EM25    060735          2996   10605#
EM26    061007          2981   10613#
EM27    061066          3071   10621#
EM28    061205          3863   10635#
EM29    061313          3351    3441   10648#
EM3     057217          9076   10450#
EM30    061361          3096   10655#
EM31    061402          2991    3091    3181    3361    3416    3461   10658#
EM32    061425          3141   10662#
EM33    061444          3171    3191    3366   10665#
EM34    061467          3186    3221   10669#
EM35    061516          3246   10673#
EM36    061544          3476    3481    3546   10677#
EM37    061607          3306   10683#
EM38    061640          3341   10688#
EM39    061713          3001    3281    3431   10696#
EM4     057272         10458#
EM40    061762          3006    3436   10703#
EM5     057346          2809    3241   10466#
EM55    062020          3346   10708#
EM6     057367          2815   10469#
EM63    062053          3421   10713#
EM65    062074          3456   10716#
EM69    062116          3591   10719#
EM7     057410          2821   10472#
EM73    062147          3231   10724#
EM74    062170          3086   10727#
EM79    062215          3236   10731#
EM8     057600         10494#
EM80    062236          2873   10734#
EM82    062275          2883   10740#
EM83    062347          2889   10748#
EM84    062404          3232    3237    3242   10753#
EM9     057654          2832   10502#
EM93    062475          3011   10763#
ERRVEC= 000004          1329#   3648    3649*   3660*   3667*   3668*   3681*   3682*   3751*   3755*   3762*   3799*   3817*
                        3818*   9235    9236*   9238*   9241*
E.A0    005404          2744#   4275*   4310*   4339*   4363*   4364    4408*   4444*   4498*   4522*   4523    4567*   4603*
                        4725*   4765*   4820*   4860*   4922*   4945*   4974*   4997*   5023*   5047*   5097*   5120*   5149*
                        5172*   5198*   5222*   5288*   5309*   5370*   5391*   5455*   5486*   5542*   5581*   5628*   5680*
                        5711*   5767*   5806*   5853*   5910*   5992*   6043*   6125*   6193*   6250*   6334*   6375*   6454*
                        6495*   6558*   6630*   7094*   7121*   7222*   7249*   7372*   7507*   7526*   7627*   7646*   7760*
                        7799*   7881*   7920*   7964*   8373*   8374*   8381*   8385    8387*   8416    9034*  11103   11113
                       11124
E.A1    005410          2746#   4277*   4312*   4341*   4364*   4365*   4410*   4446*   4500*   4523*   4524*   4569*   4605*
                        4727*   4767*   4822*   4862*   4924*   4947*   4976*   4999*   5025*   5049*   5099*   5122*   5151*
                        5174*   5200*   5224*   5290*   5311*   5372*   5393*   5457*   5488*   5544*   5583*   5630*   5682*
                        5713*   5769*   5808*   5855*   5912*   5994*   6045*   6127*   6195*   6252*   6336*   6377*   6456*
                        6497*   6560*   6632*   7096*   7123*   7224*   7251*   7374*   7509*   7528*   7629*   7648*   7762*
                        7801*   7883*   7922*   7966*   8375*   8376*   8389    8391*   8436    9036*  11103   11113   11124
E.A2    005414          2748#   4279*   4314*   4343*   4412*   4448*   4502*   4571*   4607*   4729*   4769*   4824*   4864*
                        4926*   4949*   4978*   5001*   5027*   5051*   5101*   5124*   5153*   5176*   5202*   5226*   5292*
                        5313*   5374*   5395*   5459*   5490*   5546*   5585*   5632*   5684*   5715*   5771*   5810*   5857*
```

```
                    5914* 5996* 6047* 6129* 6197* 6254* 6338* 6379* 6458* 6499* 6562* 6634* 7098*
                    7125* 7226* 7253* 7376* 7511* 7530* 7631* 7650* 7764* 7803* 7885* 7924* 7968*
                    8377* 8378* 8393  8395* 9038* 11113 11124
E.A3    005420      2750# 8379* 8380*
E.B0    005406      2745# 4276* 4311* 4340* 4366* 4409* 4445* 4499* 4525* 4568* 4604* 4726* 4766*
                    4821* 4861* 4923* 4946* 4975* 4998* 5024* 5048* 5098* 5121* 5150* 5173* 5199*
                    5223* 5289* 5310* 5371* 5392* 5456* 5487* 5543* 5582* 5629* 5681* 5712* 5768*
                    5807* 5854* 5911* 5993* 6044* 6126* 6194* 6251* 6335* 6376* 6455* 6496* 6559*
                    6631* 7095* 7122* 7223* 7250* 7373* 7508* 7527* 7628* 7647* 7761* 7800* 7882*
                    7921* 7965* 8397  8399* 8426  9035* 11103 11113 11124
E.B1    005412      2747# 4278* 4313* 4342* 4367* 4411* 4447* 4501* 4526* 4570* 4606* 4728* 4768*
                    4823* 4863* 4925* 4948* 4977* 5000* 5026* 5050* 5100* 5123* 5152* 5175* 5201*
                    5225* 5291* 5312* 5373* 5394* 5458* 5489* 5545* 5584* 5631* 5683* 5714* 5770*
                    5809* 5856* 5913* 5995* 6046* 6128* 6196* 6253* 6337* 6378* 6457* 6498* 6561*
                    6633* 7097* 7124* 7225* 7252* 7375* 7510* 7529* 7630* 7649* 7763* 7802* 7884*
                    7923* 7967* 8401  8403* 8446  9037* 11103 11113 11124
E.B2    005416      2749# 4280* 4315* 4344* 4413* 4449* 4503* 4572* 4608* 4730* 4770* 4825* 4865*
                    4927* 4950* 4979* 5002* 5028* 5052* 5102* 5125* 5154* 5177* 5203* 5227* 5293*
                    5314* 5375* 5396* 5460* 5491* 5547* 5586* 5633* 5685* 5716* 5772* 5811* 5858*
                    5915* 5997* 6048* 6130* 6198* 6255* 6339* 6380* 6459* 6500* 6563* 6635* 7099*
                    7126* 7227* 7254* 7377* 7512* 7531* 7632* 7651* 7765* 7804* 7886* 7925* 7969*
                    8405  8407* 9039* 11113 11124
E.B3    005422      2751# 4281* 4316* 4345* 4414* 4450* 4504* 4573* 4609* 4731* 4771* 4826* 4866*
                    4928* 4951* 4980* 5003* 5029* 5053* 5103* 5126* 5155* 5178* 5204* 5228* 5294*
                    5315* 5376* 5397* 5461* 5492* 5548* 5587* 5634* 5686* 5717* 5773* 5812* 5859*
                    5916* 5998* 6049* 6131* 6199* 6256* 6340* 6381* 6460* 6501* 6564* 6636* 7100*
                    7127* 7228* 7255* 7378* 7513* 7532* 7633* 7652* 7766* 7805* 7887* 7926* 7970*
                    8409  8411* 9040* 11124
E.DDT   012776      4161* 4167* 4175# 4365  4524  8381
FATT1   044006      4272  6935  7027  8277# 8997  9020
FATT2   044106      4716  4751  4811  4846  4913  4965  5088  5140  5477  5702  6330  6450  6891
                    6983  7085  7213  7955  8305#
FATT3   044214      4399  4558  6581  6653  6743  6826  8337# 8340
FC      013000      4176#
FCP1    013002      4177#
FHDHM   046222      8674# 8682
FHDTAB  046344      8707#
FLGTST  046574      8750  8754  8769#
FLOAD   046276      8691#
FMTE  = 000020      1416#
FMT1    001430      2661# 8717* 8718* 8719* 8724
FORM    047736      9050  9062#
FORMAT  001426      2660# 8717  8746
FRCYL   001344      2629# 4743* 4338* 11063 11087
FRDY    043466      3894  3908  3917  4021  4030  8182  8193  8205# 8208  8575
FRDY1   043534      8219# 8222  8561
FTITLE  001340      2624# 8073  8075*
GBA     043274      3716  8143#
GDRVS   043126      3714  8096#
GINT    043322      3718  8156#
GNS   = ****** U    1534  10186 10187 10188 10189 10190 10192 10194 10195 10196 10197 10198 10199
                    10200
GO    = 000001      1380#
GSTAT   045146      4267  4290  4656  7157  7285  7367  7489  7615  7795  7916  8288  8292  8313
                    8317  8326  8505# 8580  8591  8603  8619  8676  8693  9015
GSTAT1  045210      8371  8517#
```

```
GSTAT2  045470          8523  8532  8541  8549  8557#
GTSWR = 104406          8084  10192#
HASOF   005332          2715#  8244*  8261  11059  11068  11076  11084  11092  11100  11110  11121  11133
HBA     005322          2711#  8240*  11059  11068  11076  11084  11092  11100  11110  11121  11133
HCS1    005314          2708#  3911*  3912*  3913  3919  4024*  4025*  4026  4032  4235*  4262*  4302*  4360
                        4436*  4519  4595*  4711*  4719  4757*  4806*  4814  4852*  4908*  4916  4960*  4968
                        5015*  5083*  5091  5135*  5143  5190*  5267*  5275*  5349*  5357*  5438*  5446*  5472*
                        5480  5508  5522*  5530*  5561*  5569*  5600*  5608*  5620*  5663*  5671*  5697*  5705
                        5733  5747*  5755*  5786*  5794*  5825*  5833*  5845*  5893*  5901*  5933  5947*  5955*
                        5968*  5976*  6026*  6034*  6066  6080*  6088*  6101*  6109*  6176*  6184*  6233*  6241*
                        6285*  6293*  6305*  6313*  6321*  6353*  6361*  6405*  6413*  6425*  6433*  6441*  6473*
                        6481*  6534*  6542*  6606*  6614*  6685*  6693*  6716*  6724*  6768*  6776*  6799*  6807*
                        6866*  6874*  6887*  6888*  6889  6909*  6917*  6930*  6958*  6966*  6979*  6980*  6981
                        7001*  7009*  7022*  7058*  7066*  7080*  7088  7113*  7136*  7137*  7138  7186*  7194*
                        7208*  7216  7241*  7264*  7265*  7266  7331*  7339*  7364*  7368  7458*  7466*  7486*
                        7490  7549*  7557*  7574*  7582*  7612*  7616  7702*  7710*  7726*  7754  7781*  7784
                        7827*  7835*  7847*  7875  7902*  7905  7950*  7958  7986*  7994*  8179*  8180  8190*
                        8191  8237*  8480  8484  8508*  8557*  8558*  8559  8581  8843  8992*  9010*  9026*
                        11056  11065  11073  11081  11089  11097  11107  11118  11130
HCS2    005316          2709#  3958  3960  3964  3976  4061  4063  4067  4070  7618  8238*  8489  8493
                        11056  11065  11073  11081  11089  11097  11107  11118  11130
HDA     005324          2712#  8241*  8878  11059  11068  11076  11084  11092  11100  11110  11121  11133
HDB     005336          2717#
HDC     005334          2716#  8245*  8870  11059  11068  11076  11084  11092  11100  11110  11121  11133
HDS     005326          2713#  3962  4065  8242*  11056  11065  11073  11081  11089  11097  11107  11118  11130
HDTAB   001462          2678#  8709
HDWD    001450          2671#  11079
HD1     001424          2659#  8711*  8712*  8713*  8714*  8715*  8716*  8723
HEAD    001420          2657#  8711  8781
HEADA   001422          2658#  8663*  8664*  8665*  8666*
HER     005330          2714#  3954  4056  7493  8243*  11056  11065  11073  11081  11089  11097  11107  11118
                        11130
HMR1    005340          2718#  8246*  11056  11065  11073  11081  11089  11097  11107  11118  11130
HMR2    005342          2719#  3921  4034  4239  4268  4657  5279  5361  5450  5534  5573  5612  5675
                        5759  5798  5837  5905  5959  5980  6038  6092  6113  6188  6245  6297  6325
                        6365  6417  6445  6485  6546  6618  6697  6728  6780  6811  6878  6921  6970
                        7013  7070  7158  7198  7286  7343  7470  7561  7586  7714  7839  7998  8223*
                        8228*  8247*  8327  8525  8534  8543  8551  8604  8677  8694  9016  11056  11065
                        11073  11081  11089  11097  11107  11118  11130
HMR3    005344          2720#  8224*  8229*  8248*  8526  8535  8544  8552  8592  8620  11056  11065  11073
                        11081  11089  11097  11107  11118  11130
HOLD    043622          8209  8213  8237#  8518
HPAT    005350          2722#  8250*  11059  11068  11076  11084  11092  11100  11110  11121  11133
HPEND   005302          2690#  3743*  9005*
HPOS    005346          2721#  8249*  11059  11068  11076  11084  11092  11100  11110  11121  11133
HT    = 000011          1239#  9370  9411
HVRC  = 000400          1420#
HWC     005320          2710#  8239*  11059  11068  11076  11084  11092  11100  11110  11121  11133
HZ      001364          2638#
H.A0    005364          2733#  8416  8551*  11103  11116  11127
H.A1    005370          2735#  8436  8543*  11103  11116  11127
H.A2    005374          2737#  8534*  8630  11116  11127
H.A3    005400          2739#  8525*
H.B0    005366          2734#  8426  8552*  11103  11116  11127
H.B1    005372          2736#  8446  8544*  11103  11116  11127
H.B2    005376          2738#  8535*  8643  11116  11127
```

```
H.B3     005402          2740#   8526*   8653    8663   11127
IDAE   = 002000          1422#
IE     = 000100          1381#
ILF    = 000001          1412#
INTER    050066          8171    9116#
IOTVEC= 000020           1334#   3632*   3633*
IR     = 000100          1399#
LC       012770          4160*   4166*   4172#   7078    7206    7360    7720
LCLKF    005456          2780#   3753*
LCVEC    001330          2616#   3754
LF     = 000012          1240#   9405    9411   10205   10213   10218   10222   10232   10241   10248   10256   10263   10268
                        10272   10278   10282   10288   10294   10299   10308   10316   10323   10331   10336   10343   10348
                        10353   10358   10362   10368   10373   10377   10381   10383   10387   10392   10394   10395   10399
                        10406   10410   10417   10422   10426   10432   10450   10458   10478   10486   10549   10576   10583
                        10588   10593   10605   10626   10642   10810   10818   10847   10880   10998
LIMERR   001454          2674#
LKS      001326          2614#   3752    8891*   8906*
MASK     012772          4162*   4168*   4173#   8610    8636
MASK1    012774          4163*   4169*   4174#   8605    8621    8631    8644
MCLK   = 000400          1452#
MDS    = 001000          1402#   3958    4061    8493
MEMBAS= 172100           2620#   3670    3679
MEMERR   050024          3673    9102#
MEMPAR   010350          3667#
MEMVEC= 000114           2619#   3673*   3674*
MERD   = 001000          1453#
MESMSK= 000017           1447#
MEWD   = 002000          1454#
MIND   = 000200          1451#
MSG1     054566          8076   10205#
MSG10    056311          3852    3898   10358#
MSG11    056336          9149   10362#
MSG13    056376          3774   10368#
MSG14    056455          8359   10377#
MSG15    056500          3978    4147   10381#
MSG16    056512          4654   10383#
MSG17    056537          7425   10387#
MSG19    056567         10392#  11295
MSG19A   056576          3750*   4125*   4221*   4247*   4327*   4354*   4383*   4391*   4424*   4431*   4486*   4513*   4542*
                         4550*   4583*   4590*   4699*   4794*   4897*   4940*   4992*   5041*   5072*   5115*   5167*   5216*
                         5252*   5265*   5274*   5304*   5334*   5347*   5356*   5386*   5436*   5445*   5503*   5520*   5529*
                         5559*   5568*   5598*   5607*   5661*   5670*   5728*   5745*   5754*   5784*   5793*   5823*   5832*
                         5891*   5900*   5928*   5945*   5954*   5966*   5975*   6007*   6024*   6033*   6061*   6078*   6087*
                         6099*   6108*   6140*   6161*   6174*   6183*   6218*   6231*   6240*   6283*   6292*   6311*   6320*
                         6351*   6360*   6403*   6412*   6431*   6440*   6471*   6480*   6532*   6541*   6578*   6604*   6613*
                         6650*   6683*   6692*   6705*   6714*   6723*   6733*   6766*   6775*   6788*   6797*   6806*   6816*
                         6864*   6873*   6897*   6907*   6916*   6956*   6965*   6989*   6999*   7008*   7056*   7065*   7145*
                         7152*   7164*   7184*   7193*   7273*   7280*   7292*   7329*   7338*   7456*   7465*   7547*   7556*
                         7572*   7581*   7700*   7709*   7735*   7745*   7750*   7825*   7834*   7857*   7867*   7872*   7984*
                         7993*  10394#
MSG2     055743          3698   10316#
MSG20    056602          4467   10395#
MSG21    056625          7521    7641   10399#
MSG22    056663          4471    4630   10405#
MSG23    056667          4626   10406#
MSG3     056014          3713   10323#
```

```
MSG4     056067            3715   10331#
MSG5     056121            3717   10336#
MSG6     056167            9121   10343#
MSG7     056223            3732   10348#
MSG74    056712            8984   10410#
MSG75    056760            8986   10417#
MSG76    057015            9054   10422#
MSG77    057037            9059   10426#
MSG8     056260            4121   10353#
MSP    = 000100            1450#
MULT6    046760            8801    8827#
M.ALGN=  040000            1523#
M.DPID   050210            9164#
M.DP40   050246            9175#   9197
M.DP41   050302            9180    9188#
M.DP42   050312            9186    9192#
M.DP44   050344            9199    9205#
M.DP50   050356            9173    9210#
M.DRV  = 000007            1517#
M.HEAD=  007000            1525#   8664
M.ID   = 000003            1522#
M.PAR  = 100000            1526#   8946
M.SECT=  000760            1524#   8593    8654
M.SER  = 077770            1518#
NED    = 010000            1405#   3964    3976    4067    4070    8489
NEM    = 004000            1404#
NUDRV    012472            3990    4100#   8027
NXF    = 000004            1414#
OFFSET=  000015            1372#
OFST   = 000004            1433#
OPI    = 020000            1425#
OR     = 000200            1400#
O.ADR1   067274           11418*  11469*  11471*  11646   11647*  11685*  11700   11717   11908#
O.BACK   070020           11513   11555#
O.BD     071312           11704   11705   11853#
O.BKPT   067526           11464#  11518
O.BK1    070460           11412   11682#
O.BK2    070514           11687   11690#
O.BK3    070532           11689   11696#
O.BRK    070450           11429   11680#
O.BUF    071341           11762   11771   11776   11881#
O.B2     070614           11701   11710#
O.B3     070654           11709   11718#
O.CADV   071024           11531   11548   11571   11575   11609   11613   11718   11761#
O.CLGL   067642           11487   11489   11497#
O.CLGT=  000014           11501   11871#
O.CLS1   071250           11829   11831#
O.CR     071314           11838   11840   11841   11855#
O.CRET   067754           11508   11538#
O.CRLF   071260           11546   11607   11703   11742   11747   11838#
O.CRLS   071266           11482   11840#
O.CRS    071272           11839   11841#
O.CSR1   071310           11738*  11754   11849#
O.CSR2   071311           11739*  11755   11850#
O.CT     067276           11667*  11710*  11712*  11909#
O.C1     070426           11670#  11711
```

```
0.DCD    067572          11470   11472   11482#  11540   11595   11720
0.DCD1   067576          11483#  11532   11576
0.EFF    070102          11517   11586#
0.EFF1   070220          11598   11617#
0.ENTR   067306          11409#
0.ERR    067556          11456   11467   11479#  11502   11528   11590   11833
0.ERR1   070112          11590#  11663
0.ERR2   067732          11528#  11562   11639
0.FTYP   071166          11481   11550   11565   11611   11716   11787   11803   11808#
0.GET    071110          11435   11485   11784#  11789   11791   11793
0.GO     070264          11510   11637#
0.GO2    070342          11648#  11673
0.LG  =  000006          11439   11879#
0.LGCH   071317          11498   11859#  11871
0.LGDR   067672          11505   11507#  11520
0.LGL =  000030          11520#
0.LGL1   067644          11498#  11503
0.LGL2   067664          11499   11504#
0.MSK    067266          11591   11592   11594   11901#
0.ODT    067302           1543   11407#  11890
0.OFST   070030          11515   11561#
0.OF1    070076          11574   11576#
0.OP1    067766          11511   11544#
0.OP2    067774          11460   11546#  11557
0.ORPC   067500          11452#  11512
0.P      071307          11427*  11637*  11662   11664*  11713*  11847#
0.PR1    067264          11419   11686   11690   11900#
0.PROC   070372          11519   11662#
0.PR1    070414          11666   11668#
0.RALL   067546          11430   11465   11471#
0.RCSR=  177560          11397#  11738   11740*  11750   11752   11754*  11784
0.RDB =  177562          11396#  11786
0.REGT   067430          11435#  11509
0.RSE1   071006          11751   11754#
0.RSP    067440          11437#  11440
0.RST    067344          11408   11417#
0.RSTT   070756          11643   11669   11747#
0.RST1   067374          11416   11424#
0.RTIT   070370          11426*  11657#
0.SCAN   067602          11446   11485#  11496
0.SP     067472          11438   11447#
0.SPC    071036          11764#  11772
0.SP1    067460          11443#  11448
0.STM =  000340          11383#  11428   11642   11668   11670
0.STRT   067332          11407   11414#
0.SVR    070666          11417   11682   11724#
0.SVTT   070724          11702   11714   11738#
0.T      071306          11644*  11671*  11683   11846#
0.TBIT   070314          11644#  11684
0.TBT =  000020          11384#  11645   11672   11708
0.TC     071252          11826   11832#
0.TCLS   071234          11452   11538   11544   11555   11825#
0.TCSR=  177564          11399#  11739   11741*  11748   11755*  11808   11814
0.TDB =  177566          11398#  11810*  11816*
0.TL     071333          11436   11439   11447   11873#  11879
0.TRTC   071350          11471   11647   11886#
```

```
0.TVEC= 000014          11382#  11410   11428*  11429*
0.TYPE  071152          11706   11777   11800#  11804   11842
0.TYP1  071232          11801   11812   11820#
0.TYP2  071212          11814#  11815   11818
0.UIN   067300          11418   11646*  11685   11910#
0.UPC   067260          11411*  11641*  11656   11680*  11697   11699*  11898#
0.URO   067242          11414   11444   11891#
0.USP   067256          11415*  11725*  11726   11897#
0.UST   067262          11409*  11645*  11655   11672*  11681*  11688   11708*  11899#
0.WDS   070110          11587   11589#
0.WDS2  070126          11594#  11616
0.WDS3  070156          11605#  11618   11624   11633
0.WDS4  070214          11605   11615#
0.WRD   067722          11507   11524#
0.WRDA  067746          11525   11533#
0.WRD1  067730          11527#  11534   11551
0.WSCH  070106          11516   11588#
0.XXX   071304          11724*  11732   11845#
0.45    067406          11425   11427#
PACK  = 000003           1367#   4235
PARAM   001336           2623#   3598*   3601*   3710
PARSRT  010040           1540    3598#
PAT   = 000020           1448#
PCA   = 004000           1455#
PCD   = 010000           1456#
PCLKF   005460           2781#   3757*   3764*   8889    8904
PCVEC   001332           2617#   3758    3765
PCYL    001352           2632#
PFSRT   013004           4178#   9151
PGE   = 002000           1403#
PIP   = 020000           1441#
PIRQ  = 177772           1246#
PIRQVE= 000240           1340#
PKRB    001324           2613#
PKS     001320           2611#   3756    3763    8894*   8908*
PKSB    001322           2612#   8893*
PPTP    005432           2764#   3701*
PRGSRT  010054           3599    3602#   3865    8130
PR0   = 000000           1263#   3604    3794
PR1   = 000040           1264#
PR2   = 000100           1265#
PR3   = 000140           1266#
PR4   = 000200           1267#
PR5   = 000240           1268#   2610
PR6   = 000300           1269#   3767
PR7   = 000340           1270#   3668    3674    3818    9146    9147
PS    = 177776           1243#   1244
PSW   = 177776           1244#
PWRVEC= 000024           1335#   3638*   3639*   9136*   9145*   9146*
RCYLA   046100           8461    8643#
RCYLD   046026           8458    8630#
RDCHR = 104410           9813   10195#
RDCYLA  045756           4740    4835    8618#
RDCYLD  045672           8602#
RDDATA= 000021           1374#   7364
RDGATE= 100000           1459#
```

```
RDHEAD= 000025           1376#   7781    7902
RDLIN = 104411           8096    9887   10196#
RDOCT = 104412           8143    8156   10197#
RDSEC   045622           8590#   8799
RDTAB   004276           2684#
RDY   = 000200           1382#   8205    8219
RECAL = 000013           1371#   4262    7136    7264    9010
RESREG= 104414           9947    9999   10199#  11351
RESVEC= 000010           1330#
RHEAD   046174           8466    8663#
RHTAB   001666           2679#   7780    7814    7901    7935    8806    8808    8812    8815   11095
RKASOF= 000016           1353#   3808    7141    7269    7732    7854    8244    8281    8308    8337
RKBA  = 000004           1348#   3804    7356*   7400*   7418*   7481*   7606*   8240
RKCS1 = 000000           1346#   3801    3913*   4026*   4232*   4259*   4299*   4381*   4396*   4433*   4540*   4555*   4592*
                         4754*   4849*   5012*   5187*   5516*   5617*   5741*   5842*   5941*   6074*   6710*   6738*   6793*
                         6821*   6889*   6903*   6981*   6995*   7110*   7138*   7150*   7238*   7266*   7278*   7503*   7722*
                         7741*   7843*   7863*   8180*   8191*   8205    8219    8237    8519*   8528*   8537*   8546*   8559*
                         9007*   9023*
RKCS2 = 000010           1350#   3802    3892*   3906*   3910*   4019*   4023*   4233*   4234*   4260*   4261*   4300*   4301*
                         4434*   4435*   4593*   4594*   4755*   4756*   4850*   4851*   5013*   5014*   5188*   5189*   5262*
                         5264*   5266*   5271*   5273*   5344*   5346*   5348*   5353*   5355*   5433*   5435*   5437*   5442*
                         5444*   5517*   5519*   5521*   5526*   5528*   5556*   5558*   5560*   5565*   5567*   5595*   5597*
                         5599*   5604*   5606*   5618*   5619*   5658*   5660*   5662*   5667*   5669*   5742*   5744*   5746*
                         5751*   5753*   5781*   5783*   5785*   5790*   5792*   5820*   5822*   5824*   5829*   5831*   5843*
                         5844*   5888*   5890*   5892*   5897*   5899*   5942*   5944*   5946*   5951*   5953*   5963*   5965*
                         5967*   5972*   5974*   6021*   6023*   6025*   6030*   6032*   6075*   6077*   6079*   6084*   6086*
                         6096*   6098*   6100*   6105*   6107*   6171*   6173*   6175*   6180*   6182*   6228*   6230*   6232*
                         6237*   6239*   6280*   6282*   6284*   6289*   6291*   6308*   6310*   6312*   6317*   6319*   6348*
                         6350*   6352*   6357*   6359*   6400*   6402*   6404*   6409*   6411*   6428*   6430*   6432*   6437*
                         6439*   6468*   6470*   6472*   6477*   6479*   6529*   6531*   6533*   6538*   6540*   6601*   6603*
                         6605*   6610*   6612*   6680*   6682*   6684*   6689*   6691*   6711*   6713*   6715*   6720*   6722*
                         6763*   6765*   6767*   6772*   6774*   6794*   6796*   6798*   6803*   6805*   6861*   6863*   6865*
                         6870*   6872*   6904*   6906*   6908*   6913*   6915*   6953*   6955*   6957*   6962*   6964*   6996*
                         6998*   7000*   7005*   7007*   7053*   7055*   7057*   7062*   7064*   7111*   7112*   7181*   7183*
                         7185*   7190*   7192*   7239*   7240*   7326*   7328*   7330*   7335*   7337*   7362*   7363*   7453*
                         7455*   7457*   7462*   7464*   7483*   7544*   7546*   7548*   7553*   7555*   7569*   7571*   7573*
                         7578*   7580*   7605*   7697*   7699*   7701*   7706*   7708*   7723*   7724*   7776*   7777*   7793
                         7822*   7824*   7826*   7831*   7833*   7844*   7845*   7897*   7898*   7914    7947*   7948*   7981*
                         7983*   7985*   7990*   7992*   8238    8506*   8507*   8520*   8521*   8529*   8530*   8538*   8539*
                         8547*   8548*   8573*   8577*   8578*   9008*   9009*   9024*   9025*
RKDA  = 000006           1349#   3805    7358*   7402*   7420*   7480*   7609*   8241
RKDB  = 000024           1355#   3810    7620    7788    7789    7790    7909    7910    7911
RKDC  = 000020           1354#   3809    4709*   4746*   4804*   4841*   6304*   6424*   7078*   7206*   7360*   7482*   7607*
                         7725*   7775*   7846*   7896*   8245
RKDS  = 000012           1351#   3806    8242
RKECPS= 000030           1359#   3814    8249
RKECPT= 000032           1360#   3815    8250
RKER  = 000014           1352#   3807    8243
RKMR1 = 000026           1356#   3811    4266*   4289*   4655*   8246    8287*   8291*   8312*   8316*   8522*   8531*   8540*
                         8579*   8590*   8602*   8618*   8675*   8692*   9014*
RKMR2 = 000034           1357#   3812    8223    8228    8247
RKMR3 = 000036           1358#   3813    8224    8229    8248
RKPRI   001316           2610#   8172
RKVEC   001314           2609#   3737*   8160*   8163*   8170
RKWC  = 000002           1347#   3803    7361*   7484*   7608*   8239
RLS   = 000010           1396#   5266    5348    5437    5521    5560    5599    5662    5746    5785    5824    5892    5946
```

```
                        5967    6025    6079    6100    6175    6232    6284    6312    6352    6404    6432    6472    6533
                        6605    6684    6715    6767    6798    6865    6908    6957    7000    7057    7185    7330    7457
                        7548    7573    7701    7826    7985
RSEC    046136          8465    8653#
RTT   = 000006         11386#
SAVREG= 104413          9932    9973   10198#  11294
SBPAR   047224          8386    8390    8394    8398    8402    8406    8410    8932#
SCLR  = 000040          1398#   3892    3906    4019    8573
SCOP1 = 104415          3903    4016   10200#
SCOP1$  047276          8956#  10200
SEC     001370          2642#   7610*
SECFLG  046524          8725    8745#
SECTOR  001374          2644#   7477*   7480    7498*   7499    7609    7610    8592*   8593*   8594*   8595*   8596*   8597*
                        8653*   8654*   8655*   8656*   8657*   8658*   8800*   8804    8805*   8806*   8817    8827*   8828
                        8829*   8830*
SEEK  = 000017          1373#   4711    4806    4908    4960    5083    5135    5472    5697    6305    6425    7080    7208
                        7726    7847    7950
SELDRV= 000001          1366#   3911    4024    5267    5275    5349    5357    5438    5446    5522    5530    5561    5569
                        5600    5608    5663    5671    5747    5755    5786    5794    5825    5833    5893    5901    5947
                        5955    5968    5976    6026    6034    6080    6088    6101    6109    6176    6184    6233    6241
                        6285    6293    6313    6321    6353    6361    6405    6413    6433    6441    6473    6481    6534
                        6542    6606    6614    6685    6693    6716    6724    6768    6776    6799    6807    6866    6874
                        6909    6917    6958    6966    7001    7009    7058    7066    7186    7194    7331    7339    7458
                        7466    7549    7557    7574    7582    7702    7710    7827    7835    7986    7994    8508    8557
SETINT  043354          3738    8161    8170#
SIZFLG  005462          2782#   3739*   3896    8114*   8136*
SKI   = 000002          1413#
SORT    046652          8797#
SPACE2  067216         11329   11338   11349   11361#
SPBAR = 000040          2628#
SRTSPL= 000011          1370#   6930    7022    8992
SRTTAB  002072          2680#   8809
ST    = 177776         11380#  11409   11410*  11423*  11642*  11668*  11670*  11696*
STACK = 001100          1234#   3603    3630    3792    3843    3904    4012    4017    4115    4219    4484    4646    4697
                        4792    4895    5070    5250    5332    5432    5657    5887    6020    6159    6216    6279    6399
                        6526    6598    6677    6760    6860    6952    7052    7180    7321    7451    7692    8023    8471
                        9110    9148   11357
START   010050          1538    3601#
START1  010456          3692#
STKLMT= 177774          1245#
STOP    047330          8975#   9654    9737
ST2     010524          3699    3710#
ST3     010570          3700    3728#
ST4     010622          3712    3733    3736#
ST5     010646          3719    3741#   3889    3952    8052    9061
SUBCLR  045534          4225    4251    4331    4490    4651    4703    4798    4901    5076    5256    5338    5469    5694
                        5924    6057    6165    6222    6301    6421    6527    6550    6599    6622    6678    6701    6761
                        6784    6858    6882    6927    6950    6974    7019    7050    7074    7178    7202    7323    7348
                        7387    7474    7694    7819    7940    8573#   8977    8989
SVAL  = 100000          1443#
SWR     001140          2527#   3628    3650*   3652    3658*   3665*   8082    8422    8432    8442    8452    8956    9074
                        9082    9087    9106    9230    9244    9246    9252    9259    9297    9304    9316    9320    9657
                        9696    9751*  11353
SWREG   000176          1536#   3658    8082    9657    9696    9719
SW0   = 000001          1298#
SW00  = 000001          1288#   1298
```

```
SW01   = 000002          1287#  1297
SW02   = 000004          1286#  1296
SW03   = 000010          1285#  1295
SW04   = 000020          1284#  1294
SW05   = 000040          1283#  1293
SW06   = 000100          1282#  1292
SW07   = 000200          1281#  1291
SW08   = 000400          1280#  1290
SW09   = 001000          1279#  1289
SW1    = 000002          1297#
SW10   = 002000          1278#
SW11   = 004000          1277#
SW12   = 010000          1276#  11353
SW13   = 020000          1275#  9074
SW14   = 040000          1274#  9087
SW15   = 100000          1273#
SW2    = 000004          1296#
SW3    = 000010          1295#
SW4    = 000020          1294#
SW5    = 000040          1293#
SW6    = 000100          1292#
SW7    = 000200          1291#
SW8    = 000400          1290#
SW9    = 001000          1289#  8422   8432   8442   8452   8956   9082   9106
TBITVE= 000014           1331#
TEMP1   005352           2725#  3893*  3907*  3914*  3916*  3921*  3922*  3923   4020*  4027*  4029*  4034*  4035*
                         4036   4715*  4810*  4912*  4964*  5087*  5139*  5476*  5701*  6329*  6449*  6890*  6982*
                         7084*  7212*  7730*  7742*  7851*  7864*  7954*  8181*  8192*  8207*  8221*  8278*  8283*
                         8310*  8350   8352*  8383   8385*  8387   8389*  8391   8393*  8395   8397*  8399   8401*
                         8403   8405*  8407   8409*  8411   8413*  8505   8511*  8517   8554*  8560*  8574*  8674*
                         8679*  8691*  8696*  8727*  8730*  8731   8938*  8946*  11079
TEMP2   005354           2726#  4271*  6934*  7026*  7351*  7390*  7391   7416*  8285*  8681*  8728*  8729*  8731*
                         8732   8996*  9019*
TEMP3   005356           2727#  7352*  7393   7413*  7414
TEMP4   005360           2728#  7355*  7356   7400   7417*  7418
TEMP5   005362           2729#  7357*  7358   7401*  7402   7419*  7420
TERR1   047054           8848   8852   8868#  8873   8880
TIMER   005466           2785#  4222   4248   4328   4487   4700   4795   4898   4937   4989   5038   5073   5112
                         5164   5213   5253   5335   5643   5868   5984   6117   6162   6219
TIMUP   001372           2643#  8888*
TITLE   043046           3694   3731   8073#
TKVEC  = 000060          1338#  9634*  9635*
TMO     047204           4223   4249   4329   4488   4701   4750   4796   4845   4899   4938   4990   5039   5074
                         5113   5165   5214   5254   5336   5644   5869   5985   6118   6163   6220   6554   6626
                         6895   6987   7162   7290   8915#
TOCYL   001346           2630#  4744*  4839*  7774*  7814   7895*  7935   11063  11087  11095
TPVEC  = 000064          1339#
TRAPPC  001334           2621#  9103*  11062
TRAPVE= 000034           1337#  3636*  3637*  9147*
TRT    = 000003          11385# 11886
TRTVEC= 000014           1332#
TRUERR  047002           7495   8841#
TSTATN  043750           4305   4425   4439   4584   4598   4760   4855   5018   5042   5193   5217   5283   5305
                         5365   5387   5512   5538   5577   5623   5645   5737   5763   5802   5848   5870   5937
                         5986   6008   6070   6119   6141   6369   6489   7116   7244   8258#  9029
TST1    011044           3768   3790#
```

```
TST10   015712          4649    4695#
TST11   016352          4790#
TST12   017012          4893#
TST13   020056          5068#
TST14   021122          5248#
TST15   021534          5306    5330#
TST16   022146          5388    5430#
TST17   023556          5646    5655#
TST2    011226          3819    3841#
TST20   025166          5871    5885#
TST21   026112          6009    6018#
TST22   027036          6142    6157#
TST23   027330          6214#
TST24   027622          6277#
TST25   030472          6397#
TST26   031342          6524#
TST27   031724          6587    6596#
TST3    012076          4010#
TST30   032306          6659    6675#
TST31   032754          6749    6758#
TST32   033422          6832    6856#
TST33   034120          6948#
TST34   034616          7048#
TST35   035470          7176#
TST36   036342          7319#
TST37   037152          7406    7424    7449#
TST4    012502          4046    4113#
TST40   040562          7690#
TST5    013004          4164    4170    4217#
TST6    014554          4482#
TST7    015564          4644#
TYPDS = 104405          8041    10190#
TYPE  = 104401          3698    3713    3715    3717    3732    3774    3852    3863    3868    3898    3932    3978    4121
                        4147    4467    4471    4626    4630    4654    7425    7429    7430    7521    7641    8039    8042
                        8076    8129    8359    8984    8986    9054    9059    9076    9121    9149    9299    9307    9375
                        9469    9595    9651    9663    9717    9718    9721    9732    9742    9753    9772    9820    9826
                        9831    9835    9840    9841    9843    9846    9850    9916    9918    10058   10186#  11295   11305
                        11306   11310   11311   11320   11329   11333   11338   11340   11346   11349
TYPERR  066736          9306    11294#
TYPOC = 104402          9720    10187#  11326
TYPON = 104404          10189#
TYPOS = 104403          3871    3935    3981    4151    8362    9079    9124    10188#
T.A2  = 000001          2755#   4284    4319    4453    4612    4774    4869    5032    5207    5637    5862    7130    7258
                        7516    7535    7636    7655    7808    7929    8456    9043
T.B2  = 000002          2756#   4284    4319    4453    4612    4774    4869    5032    5207    5637    5862    7130    7258
                        7516    7535    7636    7655    7808    7929    8459    9043
T.B3  = 000004          2757#   4284    8463
T1      001376          2646#
T10     001400          2647#   3893    3907    3916    4020    4029    4271    6890    6982    8181    8560    8574    9019
T100    001406          2650#   6934    7026    8996
T50     001402          2648#
T500    001404          2649#
T5000   001410          2651#
T50000  001412          2652#   3914    4027    4715    4810    4912    4964    5087    5139    5476    5701    6329    6449
                        7084    7212    7730    7851    7954    8192
UFE   = 000400          1401#   3960    4063
```

```
UNITB   005464          2783#  3749*  4124*  4220*  4234   4244   4246*  4261   4301   4326*  4353*  4382*  4390*
                        4398   4423*  4430*  4435   4485*  4512*  4541*  4549*  4557   4582*  4589*  4594   4698*
                        4756   4793*  4851   4896*  4939*  4991*  5014   5040*  5071*  5114*  5166*  5189   5215*
                        5251*  5263*  5264   5272*  5273   5303*  5333*  5345*  5346   5354*  5355   5385*  5434*
                        5435   5443*  5444   5502*  5518*  5519   5527*  5528   5557*  5558   5566*  5567   5596*
                        5597   5605*  5606   5619   5659*  5660   5668*  5669   5727*  5743*  5744   5752*  5753
                        5782*  5783   5791*  5792   5821*  5822   5830*  5831   5844   5889*  5890   5898*  5899
                        5927*  5943*  5944   5952*  5953   5964*  5965   5973*  5974   6006*  6022*  6023   6031*
                        6032   6060*  6076*  6077   6085*  6086   6097*  6098   6106*  6107   6139*  6160*  6172*
                        6173   6181*  6182   6217*  6229*  6230   6238*  6239   6281*  6282   6290*  6291   6309*
                        6310   6318*  6319   6349*  6350   6358*  6359   6401*  6402   6410*  6411   6429*  6430
                        6438*  6439   6469*  6470   6478*  6479   6530*  6531   6539*  6540   6574*  6580   6602*
                        6603   6611*  6612   6646*  6652   6681*  6682   6690*  6691   6704*  6712*  6713   6721*
                        6722   6732*  6742   6764*  6765   6773*  6774   6787*  6795*  6796   6804*  6805   6815*
                        6825   6862*  6863   6871*  6872   6896*  6905*  6906   6914*  6915   6954*  6955   6963*
                        6964   6988*  6997*  6998   7006*  7007   7054*  7055   7063*  7064   7112   7140   7144*
                        7151*  7163*  7182*  7183   7191*  7192   7240   7268   7272*  7279*  7291*  7327*  7328
                        7336*  7337   7363   7454*  7455   7463*  7464   7545*  7546   7554*  7555   7570*  7571
                        7579*  7580   7698*  7699   7707*  7708   7724   7734*  7744*  7749*  7777   7823*  7824
                        7832*  7833   7845   7856*  7866*  7871*  7898   7948   7982*  7983   7991*  7992   8260
                        8280   8307   8374   8376   8378   8380   8507   8521   8530   8539   8548   8578   9009
                        9025
UNLD    005276          2686#  3741*  6885*  6938*  6940*  6977*  7030*  7032*  8980   9000*
UNLOAD= 000007          1369#  6887   6979
UNS   = 040000          1426#
UPE   = 020000          1406#
VERIFY  012076          3885   3890   3992#
VV    = 000100          1437#
WCE   = 040000          1407#  7618
WDCNT   001432          2662#  11079
WD1     001414          2655#  7620*  7719*  7725   7774   7775   7943*  7945   11071
WD2     001416          2656#  7621*  7720*  7846   7895   7896   7944*  7945   11071
WLE   = 004000          1423#
WORD    001446          2670#
WRDATA= 000023          1375#  7486
WRHEAD= 000027          1377#
WRL   = 004000          1440#
WRTCHK= 000031          1378#  7612
WRTGAT= 040000          1458#
$APTHD  001000          1565   1571#
$ASTAT= ****** U        9512   9527
$ATYC   051600          9483   9485#
$ATY1   051554          9481#
$ATY3   051562          9360   9482#
$ATY4   051572          9312   9484#
$AUTOB  001134          2524#  8086*  9710   9867
$BASE   001264          2594#  3736*  3800   8147*  8149*
$BDADR  001122          2519#
$BDDAT  001126          2521#
$BELL   001200          2545#  9299   9332   9663   9860
$CDW1   001270          2596#
$CDW2   001272          2597#
$CHARC  051324          9377*  9387*  9394   9403*  9408#
$CKSWR  052546          9696#  10194
$CMTAG  001100          2507#  3625   3626   3634   3640   3641   3642
$CM3  = 000000          2537#
```

```
$CM4   = 000006          2537#   2538#   2539#   2540#   2541#   2542#   2543#
$CNTLC   053456          9651    9732    9860#
$CNTLG   053470          9717    9862#
$CNTLU   053463          9742    9835    9861#
$CPUOP   001236          2568#
$CRLF    001205          2547#   3868    3932    7429    7430    9307    9332    9376    9411    9753    9840    9860    9921
                        11305   11310   11320   11333   11346
$DBLK    051544          9435    9469    9477#
$DB2D    054000          9973#   10035
$DB20    053660          7427    9932#
$DDW0    001274          2598#
$DDW1    001276          2599#
$DDW2    001300          2600#
$DDW3    001302          2601#
$DDW4    001304          2602#
$DDW5    001306          2603#
$DDW6    001310          2604#
$DDW7    001312          2605#
$DECVL   054160          9975    10021#
$DEVCT   001220          2559#   3685*   3747*   4127    8024*   8025
$DEVM    001266          2595#   3856
$DOAGN   043002          8035    8044    8050#
$DTBL    051534          9438    9473#
$ENDAD   042772          1551    8046#   9327
$ENDCT   042740          3640    8037#
$ENDMG   043011          8039    8054#
$ENULL   043006          8042    8053#
$ENV     001230          2564#   8080    9309    9355    9490    9514
$ENVM    001231          2565#   3663    3848    9357    9362    9492
$EOP     042644          3775    4243    7522    7642    8018#   11358
$EOPCT   042732          3640*   8034#   8038    9055*
$EOP1    042704          3823    4122    8026    8028#   9057    9098
$ERFLG   001103          2510#   8958    9219    9248    9250    9256*   9278    9294*   9332    11355
$ERMAX   001115          2516#   3643*   9250    9273*   9278
$ERROR   050646          3634    9292#
$ERRPC   001116          2517#   9301*   9302*   9303    9332    11055   11063   11071   11079   11087   11095   11103   11113
                        11124
$ERRTB   005470          2800#   11302
$ERTTL   001112          2514#   9300*   9332
$ESCAP   001176          2544#   3642*   3744*   4243*   4337*   4355*   4368*   4380*   4406*   4422*   4429*   4473*   4496*
                        4514*   4527*   4539*   4565*   4581*   4588*   4632*   6886*   6925*   6978*   7017*   7077*   7168*
                        7205*   7296*   8021*   8414    8418*   8428*   8438*   8448*   8468*   8472*   9056*   9102*   9105*
                        9272*   9323    9325    9332
$ETABL   001230          2563#
$ETEND   001314          1577    2606#
$FATAL   001212          2556#   9518*
$FFLG    052020          9481*   9484*   9512    9521*   9529#
$FILLC   001156          2535#   9380    9411
$FILLS   001155          2534#   9411
$GDADR   001120          2518#
$GDDAT   001124          2520#
$GET42   042762          8043#
$GTSWR   052636          9718#   10192
$HD    = 000000          1209
$HIBTS   001000          1572#
$HIOCT   053656          9909*   9920#
```

```
$ICNT   001104          2511#   9263*   9264    9266*   9277
$INTAG  001135          2525#   9715*   9734    9754    9867
$ITEMB  001114          2515#   9303*   9311    9332   11296
$LF     001206          2548#   9332    9411    9850    9860    9921
$LFLG   052017          9522*   9528#
$LPADR  001106          2512#   3644*   9085    9254*   9270*   9275    9277
$LPERR  001110          2513#   3645*   8473    8960    8963*   9111    9254    9271*   9277    9322
$MADR1  001242          2581#
$MADR2  001246          2585#
$MADR3  001252          2588#
$MADR4  001256          2591#
$MAIL   001210          1573    1577    2554#   3662    8080    9269    9309    9355
$MAMS1  001240          2575#
$MAMS2  001244          2583#
$MAMS3  001250          2586#
$MAMS4  001254          2589#
$MBADR  001002          1573#
$MFLG   052016          9482*   9488    9523*   9527#
$MNEW   053506          9721    9865#
$MSGAD  001224          2561#   9498*   9501
$MSGLG  001226          2562#   9503*
$MSGTY  001210          2555#   9496    9504*   9516    9520*
$MSWR   053475          9718    9863#
$MTYP1  001241          2576#
$MTYP2  001245          2584#
$MTYP3  001251          2587#
$MTYP4  001255          2590#
$MUL  = 000001          4465#   4467    4624#   4626
$MULT   054270          4464    4623   10077#
$MXCNT  050644          9267    9277#
$NULL   001154          2533#   9382    9411
$NWTST= 000001          3780#   3782    3826#   3828    3995#   3997    4104#   4106    4182#   4184    4476#   4478    4635#
                        4637    4679#   4681    4783#   4785    4878#   4880    5062#   5064    5237#   5239    5324#   5326
                        5406#   5408    5649#   5651    5875#   5877    6012#   6014    6145#   6147    6208#   6210    6265#
                        6267    6390#   6392    6511#   6513    6590#   6592    6662#   6664    6752#   6754    6835#   6837
                        6942#   6944    7035#   7037    7170#   7172    7299#   7301    7433#   7435    7665#   7667
$OCNT   052244          9567*   9596*   9609#
$OCTVL  053762          4658    9934    9959#
$OMODE  052246          9562*   9566*   9571    9574*   9585*   9611#
$OVER   050630          9231    9247    9255    9265    9274#
$PASS   001216          2558#   3662*   4377    4536    4648    7423    8031*   8032*   8040    8053    9261    9278
$PASTM  001006          1575#
$PWRCT  050206          9142*   9143*   9153#
$PWRDN  050122          3638    9136#   9145
$PWRUP  050134          9136    9142#
$QUES   001204          2546#   9332    9411    9772    9843    9860    9918    9921
$RDCHR  053110          9785#  10195
$RDDEC= ****** U       10198
$RDLIN  053200          9808#  10196
$RDOCT  053520          9882#  10197
$RDSZ = 000022          9801#
$RESRE  054440         10143#  10199
$RTNAD  043004          8052#
$R2A  = ****** U       10200
$SAVRE  054402         10127#  10198
$SB2D   054174          4469    4628   10033#
```

```
$SCOPE  050366          3632    9228#
$SETUP= 000137          3595#   3631    3632    3634    3636    3638    3640    3641    3642    3644    8029    8077    9229
                        9293    9319    9327    9655    9660    9661    9691    9867
$STUP = 177777          3595#
$SUPRS  054230          4470    4629    4675    7428    10050#
$SVLAD  050574          9239    9268#
$SVPC = 000244          1549#   1554
$SWR  = 167400          1194#   1209    1214    1215    1216    1217    1218    1219    1220    1221    2543    2544    2545
                        3641    3642    3644    3645    3791    3842    4011    4114    4218    4483    4645    4696    4791
                        4894    5069    5249    5331    5431    5656    5886    6019    6158    6215    6278    6398    6525
                        6597    6676    6759    6857    6949    7049    7177    7320    7450    7691    8015    8030    8045
                        8051    8053    9220    9221    9222    9223    9224    9230    9242    9244    9245    9248    9249
                        9250    9257    9258    9259    9271    9274    9277    9284    9285    9286    9287    9288    9297
                        9304    9316    9320    9332
$SWREG  001232          2566#   3665
$SWRMK= 000000          1221    1222    9224    9225    9246
$TESTN  001214          2557#   4116*   9269*   11055   11062   11063   11071   11079   11087   11095   11103   11113   11124
$TIMES  001174          2543#   3641*   3791*   3842*   4011*   4114*   4218*   4483*   4645*   4696*   4791*   4894*   5069*
                        5249*   5331*   5431*   5656*   5886*   6019*   6158*   6215*   6278*   6398*   6525*   6597*   6676*
                        6759*   6857*   6949*   7049*   7177*   7320*   7450*   7691*   8022*   8030*   9257*   9264    9267*
                        9277
$TKB    001146          2530#   9615    9636    9647    9672    9700    9727
$TKCNT  052250          9616#   9631*   9661    9678*   9792    9794*
$TKINT  052260          3608    9631#   9652    9713
$TKQEN= 052257          9620#   9686    9797
$TKQIN  052252          9617#   9632*   9633    9684*   9685*   9686    9688*
$TKQOU  052254          9618#   9633*   9795    9796*   9797    9799*
$TKQSR  052256          9619#   9632    9688    9799
$TKS    001144          2529#   9615    9637*   9668*   9670    9676*   9698    9714*   9724    9736*   9756*
$TKSRV  052330          9634    9647#
$TMP0   001160          2537#
$TMP1   001162          2538#
$TMP2   001164          2539#
$TMP3   001166          2540#
$TMP4   001170          2541#   3912    3931    3945*   3956*   4025    4040    4043*   4058*   4102*   4144*   4158    6888
                        6980    7137    7265    8179    8190    8558
$TMP5   001172          2542#   3745*   8414*   8468    8472
$TN   = 000041          1195#   1209    3768    3780    3791#   3819    3826    3842#   3995    4011#   4046    4104    4114#
                        4116    4117    4164    4170    4182    4218#   4476    4483#   4635    4645#   4649    4679    4696#
                        4783    4791#   4878    4894#   5062    5069#   5237    5249#   5306    5324    5331#   5388    5406
                        5431#   5646    5649    5656#   5871    5875    5886#   6009    6012    6019#   6142    6145    6158#
                        6208    6215#   6265    6278#   6390    6398#   6511    6525#   6587    6590    6597#   6659    6662
                        6676#   6749    6752    6759#   6832    6835    6857#   6942    6949#   7035    7049#   7170    7177#
                        7299    7320#   7406    7424    7433    7450#   7665    7691#
$TNPWR  054110          9980    9981    10001#
$TPB    001152          2532#   9400*   9411
$TPFLG  001157          2536#   9349    9411
$TPS    001150          2531#   9398    9411
$TRAP   054476          3636    10163#
$TRAP2  054520          10174#  10185
$TRP  = 000016          10178#  10187#  10188#  10189#  10190#  10191#  10192   10193#  10194   10195#  10196#  10197#  10198#
                        10199#  10200#  10201#
$TRPAD  054532          10168   10185#
$TSTM   001004          1574#
$TSTNM  001102          2509#   4117*   8029*   9219    9246    9268*   9269    9274    9278    9296    9332
$TTYIN  053434          9810    9811    9823    9841    9855    9859#
```

```
$TYPBN= ****** U        10191
$TYPDS  051330           9423#  10190
$TYPE   051046           9349#   9509   10178   10186
$TYPEC  051260           9379    9386    9393    9398#   9399    9758
$TYPEX  051326           9404    9406    9409#
$TYPOC  052046           9565#  10187
$TYPON  052062           9564    9567#  10189
$TYPOS  052022           9560#  10188
$UNIT   001222           2560#   3686*   3748*   4129*   4137    4148    4233    4260    4300    4397    4434    4556    4593
                         4755    4850    5013    5188    5262    5271    5344    5353    5433    5442    5517    5526    5556
                         5565    5595    5604    5618    5658    5667    5742    5751    5781    5790    5820    5829    5843
                         5888    5897    5942    5951    5963    5972    6021    6030    6075    6084    6096    6105    6171
                         6180    6228    6237    6280    6289    6308    6317    6348    6357    6400    6409    6428    6437
                         6468    6477    6529    6538    6579    6601    6610    6651    6680    6689    6711    6720    6741
                         6763    6772    6794    6803    6824    6861    6870    6904    6913    6953    6962    6996    7005
                         7053    7062    7111    7139    7181    7190    7239    7267    7326    7335    7362    7453    7462
                         7544    7553    7569    7578    7697    7706    7723    7731    7776    7822    7831    7844    7852
                         7897    7947    7981    7990    8259    8279    8306    8373    8375    8377    8379    8506    8520
                         8529    8538    8547    8577    9008    9024
$UNITM  001010           1576#
$USWR   001234           2567#
$VECT1  001260           2592#
$VECT2  001262           2593#
$XTSTR  050400           9233#
$$GET4= 000000           8045#
$OFILL  052245           9561*   9565*   9575    9610#
$40CAT= ****** U         9230    9306
.       = 067302         1530#   1534#   1539#   1542#   1549    1550#   1552#   1554#   1555#   1561    1562#   1564#   1566#
                         2506#   2549    2678#   2679#   2680#   2682#   2683#   2684#   3629    3644    3645    8053    8057
                         9138    9194    9277    9278    9332    9411    9477#   9530#   9615    9619#   9620    9621#   9859#
                         9860    9867#   9921    9959#  10021#  11370#  11371#  11520   11749   11753   11785   11809   11871
                        11879   11882#  11890#
.$ASTA= ****** U         9482    9485
.$X     = 001000         1561#   1566
```

CALIB	1683#	4257	9006											
CHECK	1622#	4282	4317	4346	4369	4415	4451	4505	4528	4574	4610	4732	4772	4827	4867
	4929	4952	4981	5004	5030	5054	5104	5127	5156	5179	5205	5229	5295	5316	5377
	5398	5462	5493	5549	5588	5635	5687	5718	5774	5813	5860	5917	5999	6050	6132
	6200	6257	6341	6382	6461	6502	6565	6637	7101	7128	7229	7256	7379	7514	7533
	7634	7653	7767	7806	7888	7927	7971	9041							
COMMEN	1341#														
CWD2	1637#	4289													
DRCLR	1660#	4298	4432	4591	4753	4848	5011	5186	5616	5841	7109	7237	9022		
ENDCOM	1341#														
EOPGM	1945#	8019													
ERROR	1235#	3822	3887	3895	3909	3918	3941	3950	3969	3973	3984	3988	4022	4031	4049
	4076	4080	4084	4088	4226	4230	4237	4241	4252	4255	4264	4270	4273	4285	4286
	4287	4288	4293	4296	4304	4307	4320	4321	4322	4323	4332	4336	4349	4350	4351
	4352	4358	4362	4372	4373	4374	4375	4388	4394	4400	4404	4418	4419	4420	4421
	4427	4438	4441	4454	4455	4456	4457	4491	4495	4508	4509	4510	4511	4517	4521
	4531	4532	4533	4534	4547	4553	4559	4563	4577	4578	4579	4580	4586	4597	4600
	4613	4614	4615	4616	4652	4704	4708	4713	4717	4721	4735	4736	4737	4738	4747
	4752	4759	4762	4775	4776	4777	4778	4781	4799	4803	4808	4812	4816	4830	4831
	4832	4833	4842	4847	4854	4857	4870	4871	4872	4873	4876	4902	4906	4910	4914
	4918	4932	4933	4934	4935	4942	4955	4956	4957	4958	4962	4966	4970	4984	4985
	4986	4987	4994	5007	5008	5009	5010	5017	5020	5033	5034	5035	5036	5043	5057
	5058	5059	5060	5077	5081	5085	5089	5093	5107	5108	5109	5110	5117	5130	5131
	5132	5133	5137	5141	5145	5159	5160	5161	5162	5169	5182	5183	5184	5185	5192
	5195	5208	5209	5210	5211	5218	5232	5233	5234	5235	5257	5261	5269	5277	5281
	5285	5298	5299	5300	5301	5307	5319	5320	5321	5322	5339	5343	5351	5359	5363
	5367	5380	5381	5382	5383	5389	5401	5402	5403	5404	5440	5448	5452	5465	5466
	5467	5468	5470	5474	5478	5482	5496	5497	5498	5499	5506	5510	5514	5524	5532
	5536	5539	5552	5553	5554	5555	5563	5571	5575	5578	5591	5592	5593	5594	5602
	5610	5614	5622	5625	5638	5639	5640	5641	5647	5665	5673	5677	5690	5691	5692
	5693	5695	5699	5703	5707	5721	5722	5723	5724	5731	5735	5739	5749	5757	5761
	5764	5777	5778	5779	5780	5788	5796	5800	5803	5816	5817	5818	5819	5827	5835
	5839	5847	5850	5863	5864	5865	5866	5872	5895	5903	5907	5920	5921	5922	5923
	5925	5931	5935	5939	5949	5957	5961	5970	5978	5982	5988	6002	6003	6004	6005
	6010	6028	6036	6040	6053	6054	6055	6056	6058	6064	6068	6072	6082	6090	6094
	6103	6111	6115	6121	6135	6136	6137	6138	6143	6166	6170	6178	6186	6190	6203
	6204	6205	6206	6223	6227	6235	6243	6247	6260	6261	6262	6263	6287	6295	6299
	6302	6307	6315	6323	6327	6331	6344	6345	6346	6347	6355	6363	6367	6371	6385
	6386	6387	6388	6407	6415	6419	6422	6427	6435	6443	6447	6451	6464	6465	6466
	6467	6475	6483	6487	6491	6505	6506	6507	6508	6528	6536	6544	6548	6551	6556
	6568	6569	6570	6571	6577	6582	6588	6600	6608	6616	6620	6623	6628	6640	6641
	6642	6643	6649	6654	6660	6679	6687	6695	6699	6702	6708	6718	6726	6730	6736
	6744	6750	6762	6770	6778	6782	6785	6791	6801	6809	6813	6819	6827	6833	6859
	6868	6876	6880	6883	6892	6900	6911	6919	6923	6928	6932	6936	6951	6960	6968
	6972	6975	6984	6992	7003	7011	7015	7020	7024	7028	7051	7060	7068	7072	7075
	7082	7086	7090	7104	7105	7106	7107	7115	7118	7131	7132	7133	7134	7148	7154
	7160	7166	7179	7188	7196	7200	7203	7210	7214	7218	7232	7233	7234	7235	7243
	7246	7259	7260	7261	7262	7276	7282	7288	7294	7324	7333	7341	7345	7349	7366
	7370	7382	7383	7384	7385	7388	7395	7397	7409	7460	7468	7472	7475	7488	7501
	7505	7517	7518	7519	7520	7523	7536	7537	7538	7539	7551	7559	7563	7576	7584
	7588	7614	7622	7625	7637	7638	7639	7640	7656	7657	7658	7659	7695	7704	7712
	7716	7728	7738	7746	7752	7756	7770	7771	7772	7773	7783	7786	7796	7809	7810
	7811	7812	7816	7820	7829	7837	7841	7849	7860	7868	7874	7877	7891	7892	7893
	7894	7904	7907	7917	7930	7931	7932	7933	7937	7941	7952	7956	7960	7974	7975
	7976	7977	7988	7996	8000	8486	8491	8495	8510	8524	8533	8542	8550	8576	8978
	8990	8994	8998	9012	9018	9021	9028	9031	9044	9045	9046	9047	9104		

```
ESCAPE  1341#
F.EAB   1602#  4274  4309  4338  4407  4443  4497  4566  4602  4723  4764  4818  4859  4920  4944
        4972  4996  5022  5046  5095  5119  5147  5171  5197  5221  5286  5308  5368  5390  5454
        5484  5541  5580  5627  5679  5709  5766  5805  5852  5909  5991  6042  6124  6192  6249
        6333  6373  6453  6493  6557  6629  7092  7120  7220  7248  7371  7506  7525  7626  7645
        7758  7798  7879  7919  7962  9033
GETPRI  1341#
GETSWR  1341#  8077
HDCHK3  1784#  7778  7899
HDTBL   1797#
HEADER  2032#  2044  2128  2154  2190  2211  2257  2299  2309  2338  2374  2415  2449
LOOP    1586#  3902  4015
MSG     3780#  3782  3826#  3828  3995#  3997  4104#  4106  4182#  4184  4635#  4637  4679#  4681  4878#
        4880  5237#  5239  5406#  5408  5875#  5877  6145#  6147  6265#  6267  6511#  6513  6662#  6664
        6835#  6837  7035#  7037  7299#  7301  7433#  7435  7665#  7667
MULT    1341#  4462  4621
NEWTST  1341#  3780  3826  3995  4104  4182  4476  4635  4679  4783  4878  5062  5237  5324  5406
        5649  5875  6012  6145  6208  6265  6390  6511  6590  6662  6752  6835  6942  7035  7170
        7299  7433  7665
OWNTAG  1579#  2608
POP     1341#  9464  9524  9525  9910  10106  10148
PUSH    1341#  9423  9485  9487  9508  9884  10077  10128
QKPACK  1959#  4231
QKSEEK  1809#  4710  4805  4907  4959  5082  5134  5471  5696  7079  7207  7949
QKSRT   1712#  6926  7018  8987
RDATA   1877#
RDHDR   1756#  7779  7900
RELEAS  1979#  5262  5344  5433  5517  5556  5595  5658  5742  5781  5820  5888  5942  5963  6021
        6075  6096  6171  6228  6280  6308  6348  6400  6428  6468  6529  6601  6680  6711  6763
        6794  6861  6904  6953  6996  7053  7181  7326  7453  7544  7569  7697  7822  7980
REPORT  1341#
REPTA   2025#  4785  6392
REPTST  2019#  4478  5064  5326  5651  6014  6210  6592  6754  6944  7172
SCOPE   1236#  3790  3841  4010  4113  4217  4482  4644  4695  4790  4893  5068  5248  5330  5430
        5655  5885  6018  6157  6214  6277  6397  6524  6596  6675  6758  6856  6948  7048  7176
        7319  7449  7690  8020  8028
SETP    2006#  4220  4326  4485  4698  4793  4896  5071  5251  5333  6160  6217
SETPRI  1341#  9788
SETTRA  10178#  10187  10188  10189  10190  10192  10194  10195  10196  10197  10198  10199  10200
SETUP   1341#  3624
SKIP    1341#  3768  3819  4046  4164  4170  4649  5306  5388  5646  5871  6009  6142  6587  6659
        6749  6832  7406  7424
SLASH   1341#
SPACE   1341#
STARS   1341#  1547  1558  1560  1567  2502  2549  2552  3780  3789  3826  3840  3995  4009  4104
        4112  4182  4216  4476  4481  4635  4643  4679  4694  4783  4789  4878  4892  5062  5067
        5237  5247  5324  5329  5406  5429  5649  5654  5875  5884  6012  6017  6145  6156  6208
        6213  6265  6276  6390  6396  6511  6523  6590  6595  6662  6674  6752  6757  6835  6855
        6942  6947  7035  7047  7170  7175  7299  7318  7433  7448  7665  7689  8012  9216  9280
        9334  9413  9480  9537  9614  9691  9706  9777  9801  9870  9923  9962  10024  10042  10065
        10112  10157  11284  11293
STDER1  1649#  4944  4996  5046  5119  5171  5221  6557  6629
STDER2  1654#  5308  5390  5454  5541  5580  5679  5766  5805  5909  5990  6042  6123  6192  6249
SUNIT   2039#  7362  7723  7776  7844  7897  7947
SWRSU   1341#  3646#
TRMTRP  10178#
```

```
TST10    2132#   4698    4793
TST12    2158#   4896    5071
TST14    2194#   5251    5333
TST16    2215#   5433    5658
TST20    2261#   5888    6021
TST22    2303#   6160    6217
TST24    2313#   6280    6400
TST26    2342#   6527    6599
TST30    2378#   6678    6761
TST32    2419#   6861    6953
TST32A   2453#   7053    7181
TST5     2048#   4326    4485
TYPBIN   1341#
TYPDEC   1341#   8040
TYPNAM   1341#
TYPNUM   1341#
TYPOCS   1341#   3869    3933    3979    4149    8360    9077    9122
TYPOCT   1341#   9719
TYPTXT   1341#
WDATA    1837#   7485
WRCHK    1915#   7611
WRHDR    1733#
$$CMRE   2500#
$$CMTM   2500#   2537    2538    2539    2540    2541    2542
$$ESCA   1341#
$$NEWT   1341#   3780    3826    3995    4104    4182    4476    4635    4679    4783    4878    5062    5237    5324    5406
         5649    5875    6012    6145    6208    6265    6390    6511    6590    6662    6752    6835    6942    7035    7170
         7299    7433    7665
$$SET   10178#  10187   10188   10189   10190   10192   10194   10195   10196   10197   10198   10199   10200
$$SETM   3662#
$$SKIP   1341#   3768    3819    4046    4164    4170    4649    5306    5388    5646    5871    6009    6142    6587    6659
         6749    6832    7406    7424
.EQUAT   1194#   1231
.HEADE   1194#   1199
.SETUP   1194#   3595
.SWRHI   1194#   1210
.SWRLO   1194#   1222#
.$ACT1   1194#   1545
.$APTB   2550#
.$APTH   1194#   1556
.$APTY   1194#   9478
.$CATC   1194#   1528
.$CMTA   1194#   2500
.$DB2D   1194#   9960
.$DB2O   1194#   9921
.$EOP    1194#   8010
.$ERRO   1194#   9278
.$MULT   1194#  10063
.$RDOC   1194#   9868
.$READ   1194#   9612
.$SAVE   1194#  10110
.$SB2D   1194#  10022
.$SCOP   1194#   9214
.$SUPR   1194#  10040
.$TRAP   1194#  10155
.$TYPD   1194#   9411
```

```
.$TYPE   1194#   9332
.$TYPO   1194#   9535


 ABS.  071352     000


ERRORS DETECTED:  0

DZR6GA,DZR6GA.SEQ/SOL/CRF/NL:TOC/DOC=DZR6GA.P11
RUN-TIME:  36 35 4 SECONDS
RUN-TIME RATIO:  1564/76=20.4
CORE USED:  33K  (65 PAGES)

DOCUMENT PAGES:  253
```