RM02/03/05 RM05/3/2 FCTNL TST 3 AH-F928B-MC AUG 1981
CZRMOBO FICHE 1 OF 2

RM02/03/05 RM05/3/2 FCTNL TST 3 AH-F928B-MC AUG 1981
CZRM0B0 FICHE 2 OF 2 COPYRIGHT© 80-81
MADE IN USA

```
1                                       .REM    \
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16                                                  IDENTIFICATION
17                                                  --------------
18
19                                      PRODUCT CODE:   AC-F927B-MC
20
21                                      PRODUCT NAME:   CZRMOB0  RM05/3/2 FUNCTIONAL TEST, PT 3
22
23                                      PRODUCT DATE:   APRIL 1981
24
25                                      MAINTAINER:     CX DIAGNOSTIC GROUP
26
27                                      AUTHOR:         MIKE LEAVITT
28
29
30
31                                  THE INFORMATION IN  THIS  DOCUMENT IS SUBJECT TO CHANGE  WITHOUT
32                                  NOTICE AND SHOULD  NOT BE CONSTRUED AS A COMMITMENT  BY  DIGITAL
33                                  EQUIPMENT  CORPORATION.  DIGITAL  EQUIPMENT  CORPORATION ASSUMES
34                                  NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.
35
36                                  THE SOFTWARE DESCRIBED  IN  THIS DOCUMENT  IS  FURNISHED UNDER A
37                                  LICENSE  AND  MAY  ONLY BE USED OR COPIED IN ACCORDANCE WITH THE
38                                  TERMS OF SUCH LICENSE.
39
40                                  DIGITAL EQUIPMENT  CORPORATION ASSUMES NO RESPONSIBILITY FOR THE
41                                  USE  OR  RELIABILITY OF ITS  SOFTWARE  ON  EQUIPMENT THAT IS NOT
42                                  SUPPLIED BY DIGITAL.
43
44                                       COPYRIGHT (C) 1980,1981 DIGITAL EQUIPMENT CORPORATION
```

CONTENTS

1.0 INTRODUCTION

1.1 ABSTRACT

THE RM05/3/2 SUBSYSTEM FUNCTIONAL TEST IS A STAND ALONE PROGRAM WHICH USES FUNCTIONAL MEANS TO VERIFY THE OPERABILITY OF THE RM05/3/2 DISK SUBSYSTEM. IN PARTICULAR, THE PROGRAM SERVES THE FOLLOWING PURPOSES:

TO EXPLICITLY ESTABLISH CONFIDENCE IN THE BASIC OPERATIONS OF THE DISK DRIVE, INCLUDING MECHANICAL POSITIONING AND DATA TRANSFER OPERATIONS;

TO IMPLICITLY ESTABLISH CONFIDENCE IN THE DRIVE/ADAPTER ELECTRICAL INTERFACE;

TO VERIFY THE FUNCTIONALITY OF THE RM05/3/2 SUBSYSTEM, INCLUDING THE MASSBUS CONTROLLER, MASSBUS ADAPTER AND THE DISK DRIVE.

THE TEST IS COMPRISED OF 3 PARTS, WHICH WOULD NORMALLY BE RUN IN SEQUENCE, STARTING WITH PART 1. BRIEFLY, PART 1 TESTS HOUSEKEEPING AND MECHANICAL POSITIONING OPERATIONS; PART 2 TESTS WRITE, READ AND WRITE CHECK OPERATIONS USING HEADER AND DATA; PART 3 TESTS WRITE, READ AND WRITE CHECK OPERATIONS USING DATA.

1.2 UNIT UNDER TEST

THE UNIT UNDER TEST (UUT) IS THE RM05/3/2 DISK SUBSYSTEM WHICH CONSISTS OF THE RH MASSBUS CONTROLLER, THE RM05/3/2 MASSBUS ADAPTER, AND THE STORAGE MODULE DISK DRIVE. NOTE THAT A DISK PACK IS REQUIRED FOR TESTING AND IS CONSIDERED AN INTEGRAL OF THE STORAGE MODULE DISK DRIVE.

2.0 OPERATING REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

THE FOLLOWING MINIMUM HARDWARE CONFIGURATION, ASSUMED TO BE OPERATIONAL, IS REQUIRED TO LOAD AND EXECUTE THE RM05/3/2 SUBSYSTEM FUNCTIONAL TEST:

PDP-11 PROCESSOR
20K MEMORY
KW11-L OR KW11-P CLOCK
PROGRAM LOADING DEVICE
TERMINAL
RH11 OR RH70 CONTROLLER
1 TO 8 DISK DRIVES (ANY COMBINATION OF RM05'S, RM03'S OR RM02'S)

```
                2.2  MEDIA REQUIREMENTS

     EACH UNIT BEING TESTED MUST BE LOADED WITH A  SCRATCH  DISK  PACK
BEFORE TESTING BEGINS  ON  THAT  UNIT.  THE DISK MUST BE FORMATTED AND
CONTAIN A  READABLE COPY OF THE MFG AND USR BAD SECTOR FILES.


                2.3  PREREQUISITE DIAGNOSTIC PROGRAMS

RM05/3/2 DISKLESS TEST, PART 1 & 2

RM05/3/2 FUNCTIONAL TEST, PART 1 & 2


                3.0  OPERATING PROCEDURE

                3.1  LOADING

     THE PROGRAM MAY BE LOADED BY EITHER OF THE FOLLOWING MEDIA:

     .PAPER TAPE, USING THE STANDARD PAPER TAPE LOADING PROCEDURE.
     .XXDP MEDIA, USING THE APPROPRIATE LOADING DEVICE.


                3.2  SWITCH OPTIONS

     THE FOLLOWING SWITCH OPTIONS ARE PROVIDED TO ENHANCE THE  UTILITY
OF THE PROGRAM.

SW15     HALT ON ERROR
SW14     LOOP ON TEST (CURRENTLY BEING EXECUTED)
SW13     INHIBIT ERROR TYPEOUTS
SW12     UNUSED
SW11     INHIBIT TEST ITERATIONS
SW10     BELL ON ERROR
SW09     LOOP ON ERROR
SW08     LOOP ON TEST IN SW07-00

THE LOW ORDER 8 SWITCHES (SW07-SW00), ARE  USED  IN  CONJUNCTION  WITH
SW08  TO  SPECIFY  THE OCTAL NUMBER OF THE TEST WHICH THE PROGRAM WILL
LOOP ON.


                3.3  STARTING

     THE  PROGRAM  MAY BE STARTED AT LOCATION 200 OR 204.  STARTING AT
200  WILL BE THE NORMAL STARTING ADDRESS.  STARTING AT 204 WILL ENABLE
THE  RH/RM BASE  ADDRESS TO  BE CHANGED.  IF RUNNING IN  A STAND-ALONE
ENVIRONMENT, THE PROGRAM USES CONSOLE  DIALOGUE TO  ALLOW THE OPERATOR
TO CONTROL TEST CONDITIONS.
```

3.4 HALTING

THE PROGRAM SHOULD BE HALTED BY TYPING CONTROL C FROM THE CONSOLE.

NOTE: IF THE PROGRAM IS HALTED BY ANY OTHER MEANS, BAD HEADER INFORMATION MAY BE LEFT ON THE DISK PACK. THIS OF COURSE DEPENDS ON WHICH TEST IS BEING PERFORMED AT THE TIME OF THE HALT.

3.5 RESTARTING

THE PROGRAM CAN BE RESTARTED AT ADDRESS 200 OR 204. (SEE SECTION 3.3)

4.0 OPERATOR INTERFACE

4.1 PROGRAM ID

THE PROGRAM TYPES ITS TITLE AND MAINDEC NUMBER THE FIRST TIME IT IS STARTED AFTER BEING LOADED. ALSO, A WARNING MESSAGE IS TYPED, NOTIFING THE OPERATOR OF POSSIBLE HEADER CORRUPTION IF THE PROGRAM IS HALTED IMPROPERLY. THE PROGRAM IDENTIFICATION AND THE WARNING DO NOT OCCUR IF THE PROGRAM IS RESTARTED.

4.2 CONSOLE DIALOGUE

WHEN THE PROGRAM IS RUNNING IN STAND ALONE MODE, IT ENTERS A CONSOLE DIALOGUE SEQNENCE AFTER TYPING THE PROGRAM I.D. AND WARNING MESSAGE.(SEE SECTION 4.1)

THE FIRST QUESTION TYPED OUT IS: 'TYPE HELP TEXT (L) N ?'. IF THE OPERATOR RESPONDS WITH A 'Y', THE PROGRAM WILL TYPE A BRIEF HELP MESSAGE WHICH WILL LIST SWITCH OPTIONS, ETC. ANY OTHER RESPONSE TO THE QUESTION IS CONSIDERED A 'N' AND NO HELP TEXT IS TYPED. THIS QUESTION IS ONLY ASKED ON THE INITIAL PROGRAM START AND NOT ON SUBSEQUENT START-UP'S.

ON THE PROGRAM INITIAL START AND WHEN RESTARTING AT LOCATION 204, THE OPERATOR MAY CHANGE THE RH/RM BASE ADDRESSES WITH THE FOLLOWING DIALOGUE.

EXAMPLE 1

```
RMCS1=176700 <CR>          ;NO CHANGE IN ADDRESS
RMVEC=000254 <CR>          ;NO CHANGE IN ADDRESS
```

EXAMPLE 2

```
RMCS1=176700 177200<CR> ;CHANGE BASE ADDRESS TO 177200
RMVEC=000254 260<CR>    ;CHANGE VECTOR ADDRESS TO 260
```

ON THE INITIAL START, THE NEXT QUESTION TYPED IS, 'TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S) AND TERMINATE INPUT WITH A CARRIAGE RETURN'. THEN, 'DRIVE(S):' IS TYPED AND WAITS FOR THE OPERATOR TO TYPE AN "A", TO TEST ALL POSSIBLE DRIVES OR TYPE ANY STRING OF DRIVE NUMBER(S) TO BE TESTED AND TERMINATE THE INPUT WITH A "CARRIAGE RETURN". NO COMMAS OR ANY OTHER SEPARATORS ARE NEEDED WHEN ENTERING THE DRIVE NUMBERS AS A STRING. THE PROGRAM ENTERS THE COMMA SEPARTOR AUTOMATICALLY AFTER TYPING EACH NUMBER. ON ALL SUBSEQUENT STARTS, ONLY THE 'DRIVE(S):' PROMPT IS TYPED.

THE DIAGNOSTIC THEN INITIALIZES AND REPORTS THE STATUS OF THE DRIVES WHICH WHERE PREVIOUSLY SPECIFIED FOR TESTING. THE FOLLOWING IS AN EXAMPLE PRINTOUT:

```
'UNIT STATUS:
  0    ONLINE   RM03
  1    LOAD DEVICE
  2    OFFLINE  RM05
  3    NOT PRESENT
  4    NOT PRESENT
  5    NOT AN RM05/3/2
  6    NOT PRESENT
  7    NOT PRESENT'
```

THE ABOVE UNIT STATUS SHOWS THAT DRIVE 0 WILL BE TESTED, WHILE DRIVES 1 - 7 WILL NOT BE TESTED.

THE DIAGNOSTIC THEN TYPES THE FOLLOWING MESSAGE, BASED ON THE STATUS OF THE DRIVE:

```
'DRIVE(S) TO BE TESTED, 0'
```

IF NO DRIVES ARE AVAILABLE FOR TESTING, THE FOLLOWING MESSAGE WILL BE TYPED TO THE OPERATOR:

```
'DRIVE(S) TO BE TESTED, NONE'
```

THE PROGRAM WILL THEN, EITHER START TESTING THE DRIVES AVAILABLE FOR TESTING OR RETURN TO THE BEGINNING OF THE PROGRAM AND WAIT.

ONCE THE DRIVES START TESTING, THE FOLLOWING MESSAGE WILL OCCUR AS EACH DRIVE BEGINS TO BE TESTED:

```
'DRIVE 0'
```

AFTER ALL THE DRIVES ARE COMPLETELY TESTED, THE END OF PASS MESSAGE WILL BE TYPED (SEE SECTION 4.3) AND THE PROGRAM WILL START TESTING ALL THE DRIVES AGAIN. THIS WILL CONTINUE UNTIL THE PROGRAM IS HALTED BY THE OPERATOR.

NOTE: THE LETTER LOCATED WITHIN THE BRACKETS () INDICATES THE TYPE OF RESPONSE REQUIRED BY THE USER, D=DECIMAL, O=OCTAL AND L=LETTER.

4.3 PROGRESS REPORTS

AN END OF PASS REPORT OCCURS EACH TIME THE PROGRAM IS EXECUTED FOR ALL DEVICES IN THE TEST QUE. THE END OF PASS REPORT IS AS FOLLOWS.

'END OF PASS 1'

THE FOLLOWING MESSAGE WILL ALSO OCCUR IF THERE WERE ERRORS SINCE THE LAST END OF PASS REPORT.

'TOTAL ERRORS SINCE LAST REPORT 0'

4.4 PERFORMANCE REPORT

NO PERFORMANCE REPORTS ARE GIVEN DURING THE EXECUTION OF THE PROGRAM.

4.5 PROGRAM HALTS

THERE ARE NO SCHEDULED HALTS DURING THE EXECUTION OF THE PROGRAM. PROCESSOR HALTS ARE DUE TO THE TRAP CATCHER.

4.6 ERROR REPORTS

THE FIRST LINE OF THE ERROR REPORT CONTAINS THE NUMBER OF THE UNIT (DRIVE) BEING TESTED, THE DRIVE TYPE, THE TEST NUMBER, THE ERROR NUMBER AND THE VALUE OF THE PROGRAM COUNTER WHERE THE ERROR WAS CALLED. THIS LINE IS FOLLOWED BY THE ERROR MESSAGE: ONE OR MORE LINES OF TEXT WHICH GIVE A BRIEF, YET COMPREHENSIVE DESCRIPTION OF THE ERROR. THE ERROR MESSAGE IS NORMALLY FOLLOWED BY ONE OR MORE PAIRS OF LINES CONTAINING DATA HEADERS AND DATA PERTININENT TO THE ERROR, INCLUDING EXPECTED AND ACTUAL TEST RESULTS.

THE FOLLOWING PRINTOUT SHOWS A TYPICAL ERROR MESSAGE FOR THIS PROGRAM:

```
DRV# 0 - RM03, TEST# 14, ERR# 326, PC=016654
MASSBUS DATA BUS PARITY ERROR 'MDPE'' (RMCS2, BIT 8) DETECTED
DURING WRITE COMMAND
EXPECTD     RECEVD
040300      040700
RMCS1       RMCS2   RMDS    RMER1   RMER2   RMAS
144252      040700  010700  000000  000000  000000
RMWC        RMBA    RMDA    RMOF    RMDC    RMEC1   RMEC2
177403      104604  000002  010000  000000  004066  000000
RMMR1       RMMR2   RMDT    RMSN
000010      011777  024026  177777
```

4.7 EXECUTION TIME

TIME FOR RM05:

PASS 1 OF THE PROGRAM TAKES ABOUT 15 SECONDS. PASS 2 AND

SUBSEQUENT PASSES TAKE 1 MINUTES 10 SECONDS.

TIME FOR RM02/3:

PASS 1 OF THE PROGRAM TAKES ABOUT 6 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE 40 SECONDS.

5.0 ENVIRONMENTAL SUPPORT

5.1 PROCESSOR COMPATIBILITY

THE RM05/3/2 SUBSYSTEM FUNCTIONAL TEST IS EXECUTABLE ON ANY PDP-11 PROCESSOR, PROVIDING PREVIOUSLY MENTIONED HARDWARE REQUIREMENTS ARE MET, AND PROVIDING THAT DATA THROUGHPUT ON THE SYSTEM IS SUFFICIENT TO SUSTAIN DATA TRANSFER OPERATIONS.

5.2 DUAL PORT CONFIGURATIONS

THE RM05/3/2 SUBSYSTEM FUNCTIONAL TEST DOES NOT SPECIFICALLY TEST DUAL PORT LOGIC IN THE RM05/3/2 ADAPTER BUT IS EXECUTABLE ON RM05/3/2 SUBSYSTEMS HAVING THE DUAL PORT OPTION PROVIDING THE DUAL PORT SWITCH IS SET TO THE APPROPRIATE PORT (A OR B).

5.3 MEMORY PARITY HARDWARE

MEMORY PARITY HARDWARE IS NOT USED DURING THE EXECUTION OF THE RM05/3/2 SUSBYSTEM FUNCTIONAL TEST.

5.4 MEMORY MANAGEMENT HARDWARE

MEMORY MANAGEMENT HARDWARE IS NOT USED DURING THE RM05/3/2 SUSBYSTEM FUNCTIONAL TEST. CAPABILITIES OF THE MASSBUS CONTROLLER.

5.5 ACT11, APT11 COMPATIBILITY

THE RM05/3/2 SUSBYSTEM FUNCTIONAL TEST IS COMPATIBLE WITH ACT11 AND APT11 IN BOTH DUMP AND AUTOMATIC MODES. FURTHER, THE PROGRAM WILL EXECUTE A QUICK PASS DURING THE FIRST PASS IN SUPPORT OF QUICK VERIFY MODE.

5.6 XXDP COMPATIBILITY

THE RM05/3/2 SUSBYSTEM FUNCTIONAL TEST IS COMPATIBLE WITH XXDP IN DUMP AND CHAIN MODES, AND PROVIDES MEDIA PROTECTION IN THE CASE WHERE THE RM05/3/2 IS THE XXDP LOADING DEVICE.

5.7 OPERATING SYSTEM COMPATIBILITY

THE PROGRAM IS NOT COMPATIBLE WITH ANY SOFTWARE OPERATING SYSTEM.

6.0 TEST DESCRIPTION

TEST 1 CONTROLLER ACCESS TEST

PURPOSE:

TO VERIFY THAT THE UNIBUS ADDRESS OF THE SUBSYSTEM IS CORRECT, AS DEFINED AT LOCATION $BASE.

PROCEDURE:

THE TEST TRIES TO ACCESS ALL MASSBUS CONTROLLER REGISTERS USING THE $BASE ADDRESS. REGISTER CONTENTS ARE IGNORED DURING THE TEST, AND THE TEST FAILS IF A BUS TIMEOUT OCCURS FOR ANY REGISTER TRANSFER.

IF THE TEST FAILS AND THE PROGRAM IS RUNNING IN A STAND ALONE ENVIRONMENT, I.E., LOCATION 42 IS 0, THE PROGRAM WILL JUMP TO LOCATION 204 WHICH ALLOWS THE OPERATOR TO CHANGE THE $BASE ADDRESS VIA CONSOLE DIALOGUE. OTHERWISE, THE PROGRAM ESCAPES TO THE END OF PASS HANDLER.

TEST 2 - 25 WRITE/RFAD DATA TESTS

PURPOSE:

TO 'TEST WRITE DATA AND READ DATA FUNCTIONALITY OF THE RM05/3/2 SUBSYSTEM USING A SET OF VARIABLES WHICH INCLUDE WORD COUNT, HEAD MOTION, HEAD SWITCHING AND ERROR CONDITIONS.

PROCEDURE:

ALTHOUGH EACH TEST EXERCISES A DIFFERENT VARIABLE, THE GENERAL PROCEDURE OF EACH TEST IS THE SAME. THE DRIVE IS INITIALIZED AND RECALIBRATED IF "PIP" OR "SKI" ARE ACTIVE SO THAT THERE ARE NO ERRORS WHEN A TEST BEGINS. THEN, THE TEST FORMATS THE SECTOR BEING USED, WHICH MAY VARY FROM THE PRUGRAM LISTING, BECAUSE SECTORS ARE SUBSTITUTED DURING RUN TIME IF THE SELECTED SECTOR IS LISTED IN THE BAD BLOCK TABLE OF THE LAST TRACK. FOLLOWING THAT, THE TEST PERFORMS ANY EXPLICIT SEEKS REQUIRED FOR

THE CONDITIONS OF THE TEST. REGISTERS ARE PRESET AND THE WRITE

DATA COMMAND IS EXECUTED. WHEN THE WRITE COMMAND IS COMPLETE, THE TEST STORES ALL SUBSYSTEM STATUS AND CHECKS FOR PRIMARY ERRORS WHICH PRECLUDE OTHER STATUS CHECKS. IF THERE ARE NO PRIMARY ERRORS, THE TEST VERIFIES THE RESULTS OF THE WRITE COMMAND AND THEN CHECKS FOR SECONDARY ERRORS. LOOP ADDRESSES ARE MODIFIED FOLLOWING THE SUCCESSFUL COMPLETION OF THE WRITE COMMAND IN ORDER TO SHORTEN EXECUTION TIMES AND ENHANCE SCOPING LOOPS, THEN THE PROGRAM EXECUTES THE READ DATA PORTION OF THE TEST, VERIFYING THE SAME TYPE OF ERRORS AS IN THE WRITE COMMAND.

NOTE: THE SECTOR USED DURING A TEST MAY DIFFER FROM THE PROGRAM LISTING BECAUSE THE PROGRAM SUBSTITUTES A GOOD SECTOR, IF THE ONE SELECTED IS LISTED IN THE BAD BLOCK TABLE.

TEST 2 WRITE, READ ZEROS TEST

THE TEST WRITES AND READS AN ALL ZEROS DATA FIELD, CAUSING THE DRIVE TO USE NORMAL WRITE GATE THOUGHOUT THE WRITE PROCESS.

TEST 3 WRITE, WRITE CHECK ZEROS TEST

THE TEST WRITES ANL WRITE CHECKS AN ALL ZEROS DATA FIELD, VERIFYING THAT THERE ARE .O ERRORS.

TEST 4 WRITE, WRITE CHECK ZEROS W/ WCE ERROR TEST

THE TEST WRITES AN ALL ZEROS DATA FIELD, THEN COMPLEMENTS EACH BIT IN THE LAST WORD OF THE WRITE BUFFER. A WRITE CHECK COMMAND IS EXECUTED AND THE TEST VERIFIES THAT THE CORRECT WRITE CHECK ERROR IS DETECTED.

TEST 5 WRITE, READ ONES TEST

THE TEST WRITES AND READS AN ALL ONES DATA FIELD, CAUSING THE DRIVE TO USE NORMAL WRITE GATE THOUGHOUT THE WRITE PROCESS.

TEST 6 WRITE, WRITE CHECK ONES TEST

THE TEST WRITES AND WRITE CHECKS AN ALL ONES DATA FIELD.

TEST 7 WRITE, WRITE CHECK ONES W/ WCE ERROR TEST

THE TEST WRITES AN ALL ONES DATA FIELD, THEN PERFORMS A WRITE CHECK DATA COMMAND AFTER COMPLEMENTING EACH BIT IN THE LAST WORD OF THE WRITE BUFFER. THE TEST VERIFIES THAT THE CORRECT WRITE CHECK ERROR IS DETECTED.

TEST 10 WRITE, WRITE CHECK MULTIPLE SECTORS TEST

THE TEST SEEKS TO CYLINDER 0 TO INSURE THERE IS NO HEAD MOTION DURING DATA TRANSFER. THE WRITE DATA COMMAND FOLLOWS, WITH THE WORD COUNT EQUAL TO MULTIPLE SECTORS. THE SAME SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND.

TEST 11 WRITE, READ WITH IMPLIED SEEK TEST

THIS TEST SEEKS TO THE LAST CYLINDER PRIOR TO WRITING DATA ON CYLINDER 0, TRACK 0, SECTOR 0. THE EXPLICIT SEEK INSURES THAT THERE WILL BE MAXIMUM HEAD MOTION DURING THE IMPLIED SEEK OF THE WRITE COMMAND. THE SAME OPERATION, INCLUDING THE EXPLICIT SEEK IS REPEATED FOR READ DATA.

TEST 12 WRITE, WRITE CHECK WITH HEAD SWITCHING TEST

THE TEST SEEKS TO CYLINDER 0 TO INSURE THERE IS NO HEAD MOTION DURING DATA TRANSFER. THE WRITE DATA COMMAND STARTS WITH CYLINDER 0, TRACK 0, SECTOR 31. THE WORD COUNT IS EQUAL TO MULTIPLE SECTORS WHICH CAUSES THE SUBSYSTEM TO SWITCH FROM TRACK 0 TO TRACK 1 AFTER THE FIRST OF THE MULTIPLE SECTORS ARE WRITTEN. THE SAME SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND, USING THE SAME WORD COUNT AND STARTING SECTOR.

TEST 13 WRITE, WRITE CHECK WITH MID-TRANSFER SEEK TEST

THIS TEST WRITES MULTIPLE SECTORS STARTING WITH CYLINDER 0, LAST TRACK AND SECTOR 31., CAUSING A MID-TRANSFER SEEK AFTER THE FIRST SECTOR IS WRITTEN. THE SAME SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND.

TEST 14 WRITE, READ W/ HCE ERROR TEST

A SECTOR IS FORMATTED WITH AN INCORRECT HEADER, THEN THE TEST WRITES AND READS DATA FROM THE SECTOR AND VERIFIES THAT THE CORRECT ERROR IS DETECTED. EACH BIT POSITION OF BOTH HEADER WORDS IS TESTED IN THIS MANNER.

TEST 15 WRITE, READ W/ HCI TEST

A SECTOR IS FORMATTED WITH AN INCORRECT HEADER, THEN WRITTEN AND READ WITH HEADER COMPARE INHIBITED. THE TEST VERIFIES THAT NO ERROR IS DETECTED.

TEST 16 WRITE, READ W/ IVC ERROR TEST

VOLUME VALID IS RESET BY SETTING AND RESETTING DIAGNOSTIC MODE. THE TEST THEN EXECUTES A WRITE DATA COMMAND AND VERIFIES THAT INVALID COMMAND STATUS SETS. THE TEST IS REPEATED FOR READ DATA COMMAND.

TEST 17 WRITE, READ W/ ABORT TEST

THE TEST SETS AN ERROR IN THE ERROR REGISTER AND EXECUTES A WRITE DATA COMMAND VERIFYING THAT 'PIP' REMAINS INACTIVE. THE SAME PROCEDURE IS USED FOR READ DATA COMMAND.

TEST 20 WRITE, READ EACH CURRENT LEVEL TEST

THE TEST WRITES AND READS ON EACH OF THE FOLLOWING CYLINDERS IN ORDER TO WRITE AT EACH POSSIBLE CURRENT THRESHOLD.

CYLINDER	0	110 MA
"	128	104 MA
"	256	97 MA
"	384	91 MA
"	512	84 MA
"	640	77 MA
"	768	70 MA

TEST 21 WRITE, WRITE CHECK W/ CURRENT LEVEL SWITCHING TEST

THE TEST WRITES 2 SECTORS STARTING WITH THE LAST SECTOR OF CYLINDER 383. THE FIRST SECTOR IS WRITTEN AT ONE CURRENT LEVEL AND THE SECOND SECTOR IS WRITTEN AT ANOTHER CURRENT LEVEL. BOTH SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND.

TEST 22 WRITE, READ EARLY PEAK SHIFT TEST

THIS TEST WRITES AND READS A SINGLE SECTOR USING A DATA PATTERN DESIGNED TO INTRODUCE EARLY PEAK SHIFT.

TEST 23 WRITE, READ MIXED PEAK SHIFT TEST

THIS TEST WRITES AND READS A SINGLE SECTOR USING A DATA PATTERN DESIGNED TO INTRODUCE MIXED PEAK SHIFT.

TEST 24 WRITE, READ EACH TRACK TEST

THE TEST WRITES AND READS WITH EACH HEAD USING A MIX OF EARLY, NORMAL AND LATE WRITE GATES.

TEST 25 READ, WRITES CHECK MULTIPLE SECTORS IN OFFSET MODE TEST

THE TEST EXECUTES READ DATA AND WRITE CHECK DATA COMMANDS IN OFFSET MODE IN BOTH OFFSET DIRECTIONS WITH THE TRANSFER SIZE OF TWO SECTORS AT A TIME.

628

```
                        ;*LAST REVISION 04-APR-81
479
480                     .TITLE  CZRMOBO RM05/3/2 FCTNL TST 3
                        ;*COPYRIGHT (C) 1981
                        ;*DIGITAL EQUIPMENT CORPORATION
                        ;*COLORADO SPGS., CO. 80919
                        ;*
                        ;*PROGRAM BY MIKE LEAVITT
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C5), 18-MAR-81
                        ;*
481                     .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            UNUSED
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
482                     ;*         7            TN128
                        ;*         6            TN64
                        ;*         5            TN32
                        ;*         4            TN16
                        ;*         3            TN8
                        ;*         2            TN4
                        ;*         1            TN2
                        ;*         0            TN1
483
484                     .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK   = 1100
104000                  ERROR   = EMT           ;;BASIC DEFINITION OF ERROR CALL
000004                  SCOPE   = IOT           ;;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
177776                  PSW=PS
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
001000                  SW9=SW09
000400                  SW8=SW08
000200                  SW7=SW07
000100                  SW6=SW06
000040                  SW5=SW05
000020                  SW4=SW04
000010                  SW3=SW03
000004                  SW2=SW02
000002                  SW1-SW01
000001                  SW0=SW00

                        ;*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
```

```
                        BIT04   = 20
                        BIT03   = 10
                        BIT02   = 4
                        BIT01   = 2
                        BIT00   = 1
                        BIT9=BIT09
                        BIT8=BIT08
                        BIT7=BIT07
                        BIT6=BIT06
                        BIT5=BIT05
                        BIT4=BIT04
                        BIT3=BIT03
                        BIT2=BIT02
                        BIT1=BIT01
                        BIT0=BIT00

                        ;*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
485
486                     .SBTTL  RM REGISTER BIT DEFINITIONS
487
488                     ;*RMCS1 CONTROL STATUS REGISTER
489
490     004000          DVA     = BIT11                 ;DEVICE AVAILABLE-READ ONLY
491     000040          F4      = BIT05                 ;FUNCTION CODE
492     000020          F3      = BIT04                 ;FUNCTION CODE
493     000010          F2      = BIT03                 ;FUNCTION CODE
494     000004          F1      = BIT02                 ;FUNCTION CODE
495     000002          F0      = BIT01                 ;FUNCTION CODE
496     000001          GO      = BIT00                 ;GO BIT
497     000077          FNCMSK  = 000077                ;FUNCTION CODE MASK
498
499                     ;FUNCTION CODES (BITS 01-05 OF RMCS1)
500     000000          NOP     = 000000                ;NOP COMMAND
501     000002          ILF02   = 000002                ;ILLEGAL COMMAND
502     000004          SEEK    = 000004                ;SEEK COMMAND
503     000006          RECAL   = 000006                ;RECALIBRATE COMMAND
504     000010          DRVCLR  = 000010                ;DRIVE CLEAR COMMAND
505     000012          RLEASE  = 000012                ;RELEASE COMMAND
506     000014          OFFSET  = 000014                ;OFFSET COMMAND
507     000016          RTC     = 000016                ;RETURN TO CENTERLINE COMMAND
508     000020          RIP     = 000020                ;READ IN PRESET COMMAND
509     000022          PAKACK  = 000022                ;PACK ACKNOWLEDGE COMMAND
510     000022          PACACK  = PAKACK
511     000024          ILF24   = 000024                ;ILLEGAL COMMAND
512     000026          ILF26   = 000026                ;ILLEGAL COMMAND
```

```
                        SEARCH  - 000030                ;SEARCH COMMAND
                        ILF30   = 000030                ;ILLEGAL COMMAND
                        ILF32   - 000032                ;ILLEGAL COMMAND
                        ILF34   - 000034                ;ILLEGAL COMMAND
                        ILF36   = 000036                ;ILLEGAL COMMAND
                        ILF40   = 000040                ;ILLEGAL COMMAND
                        ILF42   - 000042                ;ILLEGAL COMMAND
                        ILF44   - 000044                ;ILLEGAL COMMAND
                        ILF46   = 000046                ;ILLEGAL COMMAND
                        WCD     = 000050                ;WRITE CHECK DATA COMMAND
                        WCH     - 000052                ;WRITE CHECK HEADER AND DATA
                        ILF54   = 000054                ;ILLEGAL COMMAND
                        ILF56   - 000056                ;ILLEGAL COMMAND
                        WD      = 000060                ;WRITE DATA COMMAND
                        WH      = 000062                ;WRITE HEADER AND DATA COMMAND
                        ILF64   = 000064                ;ILLEGAL COMMAND
                        ILF66   = 000066                ;ILLEGAL COMMAND
                        RD      = 000070                ;READ DATA COMMAND
                        RH      = 000072                ;READ HEADER AND DATA COMMAND
                        ILF74   = 000074                ;ILLEGAL COMMAND
                        ILF76   = 000076                ;ILLEGAL COMMAND

                        ;*RMDA  DISK ADDRESS REGISTER

                        ;TRACK ADDRESS DEFINITIONS
            010000      TA16    = BIT12                 ;TRACK ADDRESS 16.
            004000      TA8     = BIT11                 ;TRACK ADDRESS 8.
            002000      TA4     = BIT10                 ;TRACK ADDRESS 4
            001000      TA2     = BIT09                 ;TRACK ADDRESS 2
            000400      TA1     = BIT08                 ;TRACK ADDRESS 1

                        ;SECTOR ADDRESS DEFINITIONS
            000020      SA16    = BIT04                 ;SECTOR ADDRESS 16.
            000010      SA8     = BIT03                 ;SECTOR ADDRESS 8.
            000004      SA4     = BIT02                 ;SECTOR ADDRESS 4
            000002      SA2     = BIT01                 ;SECTOR ADDRESS 2
            000001      SA1     = BIT00                 ;SECTOR ADDRESS 1

                        ;TRACK & SECTOR MASKS
            177400      TADMSK  = 177400                ;TRACK ADDRESS MASK
            000377      SADMSK  = 000377                ;SECTOR ADDRESS MASK

                        ;*RMDS  DRIVE STATUS REGISTER

            100000      ATA     = BIT15                 ;ATTENTION ACTIVE
            040000      ERR     - BIT14                 ;COMPOSITE ERROR
            020000      PIP     = BIT13                 ;POSITIONING IN PROGRESS
            010000      MOL     = BIT12                 ;MEDIUM ON LINE
            004000      WRL     = BIT11                 ;WRITE LOCK
            002000      LBT     = BIT10                 ;LAST BLOCK TRANSFERRED
            001000      PGM     = BIT09                 ;PROGRAMMABLE
            000400      DPR       BIT08                 ;DRIVE PRESENT
            000200      DRY     = BIT07                 ;DRIVE READY
            000100      VV      - BIT06                 ;VOLUME VALID
            000001      OM        BIT00                 ;OFFSET MODE ACTIVE

                        ;*RMER1 ERROR REGISTER #1
```

```
                        DCK     = BIT15                 ;DATA CHECK ERROR
                        UNS     = BIT14                 ;DRIVE UNSAFE
                        OPI     = BIT13                 ;OPERATION INCOMPLETE
                        DTE     = BIT12                 ;DRIVE TIMING ERROR
                        WLE     = BIT11                 ;WRITE LOCK ERROR
                        IAE     = BIT10                 ;INVALID ADDRESS ERROR
                        AOE     = BIT09                 ;ADDRESS OVERFLOW ERROR
                        HCRC    = BIT08                 ;HEADER CRC ERROR
                        HCE     = BIT07                 ;HEADER COMPARE ERROR
                        ECH     = BIT06                 ;ECC "HARD" ERROR
                        WCF     = BIT05                 ;WRITE CLOCK FAILURE
                        FER     = BIT04                 ;FORMAT ERROR
                        PAR     = BIT03                 ;PARITY ERROR
                        RMR     = BIT02                 ;REGISTER MODIFICATION REFUSED
                        ILR     = BIT01                 ;ILLEGAL REGISTER
                        ILF     = BIT00                 ;ILLEGAL FUNCTION

                NDTMSK  = DCK.DTE!WLE!AOE!HCRC!HCE!ECH!WCF!FER
                ;"NDTMSK" IS USED TO MASK ERROR REGISTER 1 DURING NON - DATA
                ;COMMANDS, I.E., HOUSEKEEPING AND POSITIONING COMMANDS

                ;*RMAS  ATTENTION SUMMARY REGISTER

                ATNMSK  = 377                           ;MASK FOR ATTENTION BITS

                ;*RMLA  LOOK AHEAD REGISTER

                SC4     = BIT10                         ;SECTOR COUNT = 16
                SC3     = BIT09                         ;SECTOR COUNT = 8
                SC2     = BIT08                         ;SECTOR COUNT = 4
                SC1     = BIT07                         ;SECTOR COUNT = 2
                SC0     = BIT06                         ;SECTOR COUNT = 1

                SCTMSK  = 003700                        ;SECTOR COUNT MASK

                ;*RMMR1 MAINTENANCE REGISTER #1

                ;WRITE ONLY BITS
                DBCK    = BIT15                         ;DEBUG CLOCK
                DBEN    = BIT14                         ;DEBUG CLOCK ENABLE
                DEBL    = BIT13                         ;DIAGNOSTIC END OF BLOCK
                DTO     = BIT12                         ;DIAGNOSTIC TIMEOUT
                MCLK    = BIT11                         ;MAINTENANCE CLOCK
                MRD     = BIT10                         ;READ DATA
                MUR     = BIT09                         ;UNIT READY
                MOC     = BIT08                         ;ON CYLINDER
                MSER    = BIT07                         ;SEEK ERROR
                MDF     = BIT06                         ;DRIVE FAULT
                MS      = BIT05                         ;SECTOR PULSE
                MWP     = BIT03                         ;WRITE PROTECT
                MI      = BIT02                         ;INDEX PULSE
                MSC     = BIT01                         ;SECTOR COMPARE
                DMD     = BIT00                         ;DIAGNOSTIC MODE

                ;READ ONLY BITS
                OCC     = BIT15                         ;OCCUPIED
```

```
                040000          RG      = BIT14                 ;RUN AND GO
                020000          EBL     = BIT13                 ;END OF BLOCK
                010000          REX     = BIT12                 ;EXCEPTION
                004000          ESRC    = BIT11                 ;ENABLE SEARCH
                002000          PLFS    = BIT10                 ;LOOKING FOR SYNC
                001000          ECRC    = BIT09                 ;ENABLE CRC OUT
                000400          PDA     = BIT08                 ;DATA AREA
                000200          PHA     = BIT07                 ;HEADER AREA
                000100          CONT    = BIT06                 ;CONTINUE
                000040          WC      = BIT05                 ;WORD CLOCK
                000020          EECC    = BIT04                 ;ENABLE ECC OUT
                000010          MWD     = BIT03                 ;WRITE DATA BIT
                000004          LS      = BIT02                 ;LAST SECTOR
                000002          LST     = BIT01                 ;LAST SECTOR AND TRACK
                000001          DMD     = BIT00                 ;DIAGNOSTIC MODE

                                ;*RMDT  DRIVE TYPE REGISTER

                100000          NSA     = BIT15                 ;NOT SECTOR ADDRESSED = 0
                040000          TAP     = BIT14                 ;TAPE DRIVE = 0
                020000          MOH     = BIT13                 ;MOVING HEAD = 1
                004000          DRQ     = BIT11                 ;DRIVE REQUEST REQUIRED

                020024          SNGPRT  = 020024                ;SINGLE PORT DRIVE TYPE (RM02)
                024024          DULPRT  = 024024                ;DUAL PORT DRIVE TYPE (RM02)

                                ;*RMOF  OFFSET REGISTER

                010000          FMT16   = BIT12                 ;16 BIT WORD FORMAT
                004000          ECI     = BIT11                 ;ECC INHIBIT
                002000          HCI     = BIT10                 ;HEADER COMPARE INHIBIT
                000200          OFD     = BIT07                 ;OFFSET FORWARD

                                ;*RMDC  DESIRED CYLINDER ADDRESS REGISTER

                001777          CYLMSK  = 001777                ;MASK FOR CYLINDER ADDRESS

                                ;*RMMR2 MAINTENANCE REGISTER #2

                                ;READ ONLY BITS
                100000          RQA     = BIT15                 ;PORT A REQUEST
                040000          RQB     = BIT14                 ;PORT B REQUEST
                020000          TAG     = BIT13                 ;TAG CONTROL
                010000          TST     = BIT12                 ;COMMAND SEQUENCE TEST BIT
                004000          CC      = BIT11                 ;CONTROL OR CYLINDER TAG
                002000          CH      = BIT10                 ;CONTROL OR HEAD TAG
                001000          BB09    = BIT09                 ;TAG BUS
                000400          BB08    = BIT08                 ;TAG BUS
                000200          BB07    = BIT07                 ;TAG BUS
                000100          BB06    = BIT06                 ;TAG BUS
                000040          BB05    = BIT05                 ;TAG BUS
                000020          BB04    = BIT04                 ;TAG BUS
                000010          BB03    = BIT03                 ;TAG BUS
                000004          BB02    = BIT02                 ;TAG BUS
                000002          BB01    = BIT01                 ;TAG BUS
                000001          BB00    = BIT00                 ;TAG BUS
```

```
                        ;*RMER2 ERROR REGISTER 2

100000          BSE     = BIT15         ;BAD SECTOR ERROR
040000          SKI     = BIT14         ;SEEK INCOMPLETE
020000          OPE     = BIT13         ;OPERATOR PLUG FRROR
010000          IVC     = BIT12         ;INVALID COMMAND ERROR
004000          LSC     = BIT11         ;LOSS OF SYSTEM CLOCK
002000          LBC     = BIT10         ;LOSS OF BIT CLOCK
000200          DVC     = BIT07         ;DEVICE CHECK
000010          DPE     = BIT03         ;DATA PARITY ERROR

                .SBTTL  PROGRAM MNEMONICS

100000          MSE     = BIT15         ;MANUFACTURING DETECTED SECTOR ERROR
040000          USE     = BIT14         ;USER DETECTED SECTOR ERROR

                .SBTTL  RM REGISTER INDEX VALUES

000000          RMCS1   = 00            ;CONTROL STATUS REGISTER #1
000006          RMDA    = 06            ;DISK ADDRESS REGISTER
000012          RMDS    = 12            ;DRIVE STATUS REGISTER
000014          RMER1   = 14            ;ERROR REGISTER #1
000016          RMAS    = 16            ;ATTENTION SUMMARY REGISTER
000020          RMLA    = 20            ;LOOK AHEAD REGISTER
000024          RMMR1   = 24            ;MAINTENANCE REGISTER
000026          RMDT    = 26            ;DRIVE TYPE REGISTER
000030          RMSN    = 30            ;SERIAL NUMBER REGISTER
000032          RMOF    = 32            ;OFFSET REGISTER
000034          RMDC    = 34            ;DESIRED CYLINDER REGISTER
000036          RMHR    = 36            ;HOLDING REGISTER
000040          RMMR2   = 40            ;MAINTENANCE REGISTER #2
000042          RMER2   = 42            ;ERROR REGISTER #2
000044          RMEC1   = 44            ;ECC POSITION REGISTER
000046          RMEC2   = 46            ;ECC PATTERN REGISTER
000050          ILRG50  = 50            ;ILLEGAL REGISTER 50
000052          ILRG52  = 52            ;ILLEGAL REGISTER 52
000054          ILRG54  = 54            ;ILLEGAL REGISTER 54
000056          ILRG56  = 56            ;ILLEGAL REGISTER 56
000060          ILRG60  = 60            ;ILLEGAL REGISTER 60
000062          ILRG62  = 62            ;ILLEGAL REGISTER 62
000064          ILRG64  = 64            ;ILLEGAL REGISTER 64
000066          ILRG66  = 66            ;ILLEGAL REGISTER 66
000070          ILRG70  = 70            ;ILLEGAL REGISTER 70
000072          ILRG72  = 72            ;ILLEGAL REGISTER 72
000074          ILRG74  = 74            ;ILLEGAL REGISTER 74
000076          ILRG76  = 76            ;ILLEGAL REGISTER 76

000077          IDXMSK  = 77            ;MASK FOR REGISTER INDEX NUMBER

                .SBTTL  RH CONTROLLER REGISTER BIT DEFINITIONS

                ;*RMCS1 CONTROL STATUS REGISTER #1

100000          SC      = BIT15         ;SPECIAL CONDITION-READ ONLY
040000          TRE     = BIT14         ;TRANSFER ERROR
020000          MCPE    = BIT13         ;MASSBUS CONTROL BUS PARITY ERROR-READ ONLY
002000          PSEL    = BIT10         ;PORT B SELECT
```

```
                001000          A17     = BIT09                 ;ADDRESS EXTENSION
                000400          A16     = BIT08                 ;ADDRESS EXTENSION
                000200          RDY     = BIT07                 ;READY-READ ONLY
                000100          IE      = BIT06                 ;INTERRUPT ENABLE

                                ;*RMCS2 RH CONTROL STATUS REGISTER #2

                100000          DLT     = BIT15                 ;DATA LATE-READ ONLY
                040000          WCE     = BIT14                 ;WRITE CHECK ERROR-READ ONLY
                020000          UPE     = BIT13                 ;UNIBUS PARITY ERROR
                010000          NED     = BIT12                 ;NONEXISTANT DRIVE-READ ONLY
                004000          NEM     = BIT11                 ;NONEXISTANT MEMORY-READ ONLY
                002000          PGE     = BIT10                 ;PROGRAM ERROR-READ ONLY
                001000          MXF     = BIT09                 ;MISSED TRANSFER
                000400          MDPE    = BIT08                 ;MASSBUS DATA BUS PARITY ERROR-READ ONLY
                000200          OR      = BIT07                 ;OUTPUT READY-READ ONLY
                000100          IR      = BIT06                 ;INPUT READY-READ ONLY
                000040          CLR     = BIT05                 ;CONTROLLER CLEAR
                000020          PAT     = BIT04                 ;PARITY TEST
                000010          BAI     = BIT03                 ;UNIBUS ADDRESS INCREMENT INHIBIT
                000004          U2      = BIT02                 ;UNIT SELECT
                000002          U1      = BIT01                 ;UNIT SELECT
                000001          U0      = BIT00                 ;UNIT SELECT

                                ;UNIT SELECT MASK

                000007          UNTMSK  = 7                     ;UNIT SELECT MASK

                                ;*RMCS3 RH70 CONTROL STATUS REGISTER #3

                100000          APE     = BIT15                 ;ADDRESS PARITY ERROR
                040000          DPEHI   = BIT14                 ;DATA PARITY ERROR HIGH WORD
                020000          DPELO   = BIT13                 ;DATA PARITY ERROR LOW WORD
                010000          WCEHI   = BIT12                 ;WRITE CHECK ERROR HIGH WORD
                004000          WCELO   = BIT11                 ;WRITE CHECK ERROR LOW WORD
                002000          DBL     = BIT10                 ;DOUBLE WORD TRANSFER
                000100          IE      = BIT06                 ;INTERRUPT ENABLE
                000010          IPCK3   = BIT03                 ;INVERT PARITY CHECK
                000004          IPCK2   = BIT02                 ;INVERT PARITY CHECK
                000002          IPCK1   = BIT01                 ;INVERT PARITY CHECK
                000001          IPCK0   = BIT00                 ;INVERT PARITY CHECK

                                .SBTTL  RH CONTROLLER REGISTER INDEX VALUES

                000000          RMCS1   = 00                    ;CONTROL, STATUS REGISTER #1
                000002          RMWC    = 02                    ;WORD COUNT REGISTER
                000004          RMBA    = 04                    ;BUS ADDRESS REGISTER
                000010          RMCS2   = 10                    ;CONTROL, STATUS REGISTER #2
                000022          RMDB    = 22                    ;DATA BUFFER
                000050          RMBAE   = 50                    ;BUS ADDRESS EXTENSION
                000052          RMCS3   = 52                    ;CONTROL, STATUS REGISTER #3

                176700          ABASE   = 176700                ;UNIBUS ADDRESS
                120254          AVECT1  = 120254                ;UNIBUS VECTOR ADDRESS AND PRIORITY
```

```
                                .SBTTL  TRAP CATCHER

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

                                .SBTTL  STARTING ADDRESS(ES)

  000200  000137  005432       START:  JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
2
3 000204  000137  005422       START1: JMP     @#START1        ;CHANGE RH/RM BUS ADDRESS
4
5                               .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000210          $SVPC=.                 ;SAVE PC
                        000046          .=46
  000046  037242                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
  000052  000000                        .WORD   0               ;;2)SET LOC.52 TO ZERO
                        000210          .=$SVPC                 ;; RESTORE PC
6
7               001100          .=1100
8                               .SBTTL  APT PARAMETER BLOCK

                                ;;**************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**************************************************************
                        001100          .$X=.   ;;SAVE CURRENT LOCATION
                        000024          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
  000024  000200                        200     ;;FOR APT START UP
                        000044          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
  000044  001100                        $APTHDR ;;POINT TO APT HEADER BLOCK
                        001100          .=.$X   ;;RESET LOCATION COUNTER
                                ;;**************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

  001100                        $APTHD:
  001100  000000                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
  001102  001222                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
  001104  000006                $TSTM:  .WORD   6       ;;RUN TIM OF LONGEST TEST
  001106  000006                $PASTM: .WORD   6       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
  001110  000006                $UNITM: .WORD   6       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDED UNIT
  001112  000042                        .WORD   $ETEND-$MAIL/2  ;;LENGTH MAILBOX-ETABLE(WORDS)
9               001114          TAGADR=.
```

```
      0                                 .SBTTL  COMMON TAGS

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

            001114                              .=TAGADR
  001114                        $CMTAG:                         ;;START OF COMMON TAGS
  001114    000000                      .WORD   0
  001116       000              $TSTNM: .BYTE   0                ;;CONTAINS THE TEST NUMBER
  001117       000              $ERFLG: .BYTE   0                ;;CONTAINS ERROR FLAG
  001120    000000              $ICNT:  .WORD   0                ;;CONTAINS SUBTEST ITERATION COUNT
  001122    000000              $LPADR: .WORD   0                ;;CONTAINS SCOPE LOOP ADDRESS
  001124    000000              $LPERR: .WORD   0                ;;CONTAINS SCOPE RETURN FOR ERRORS
  001126    000000              $ERTTL: .WORD   0                ;;CONTAINS TOTAL ERRORS DETECTED
  001130       000              $ITEMB: .BYTE   0                ;;CONTAINS ITEM CONTROL BYTE
  001131       001              $ERMAX: .BYTE   1                ;;CONTAINS MAX. ERRORS PER TEST
  001132    000000              $ERRPC: .WORD   0                ;;CONTAINS PC OF LAST ERROR INSTRUCTION
  001134    000000              $GDADR: .WORD   0                ;;CONTAINS ADDRESS OF 'GOOD' DATA
  001136    000000              $BDADR: .WORD   0                ;;CONTAINS ADDRESS OF 'BAD' DATA
  001140    000000              $GDDAT: .WORD   0                ;;CONTAINS 'GOOD' DATA
  001142    000000              $BDDAT: .WORD   0                ;;CONTAINS 'BAD' DATA
  001144    000000                      .WORD   0                ;;RESERVED--NOT TO BE USED
  001146    000000                      .WORD   0
  001150       000              $AUTOB: .BYTE   0                ;;AUTOMATIC MODE INDICATOR
  001151       000              $INTAG: .BYTE   0                ;;INTERRUPT MODE INDICATOR
  001152    000000                      .WORD   0
  001154    177570              SWR:    .WORD   DSWR             ;;ADDRESS OF SWITCH REGISTER
  001156    177570              DISPLAY: .WORD  DDISP            ;;ADDRESS OF DISPLAY REGISTER
  001160    177560              $TKS:   177560                   ;;TTY KBD STATUS
  001162    177562              $TKB:   177562                   ;;TTY KBD BUFFER
  001164    177564              $TPS:   177564                   ;;TTY PRINTER STATUS REG. ADDRESS
  001166    177566              $TPB:   177566                   ;;TTY PRINTER BUFFER REG. ADDRESS
  001170       000              $NULL:  .BYTE   0                ;;CONTAINS NULL CHARACTER FOR FILLS
  001171       002              $FILLS: .BYTE   2                ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
  001172       012              $FILLC: .BYTE   12               ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
  001173       000              $TPFLG: .BYTE   0                ;;''TERMINAL AVAILABLE'' FLAG (BIT<07>=0-YES)
  001174    000000              $TMP0:  .WORD   0                ;;USER DEFINED
  001176    000000              $TMP1:  .WORD   0                ;;USER DEFINED
  001200    000000              $TMP2:  .WORD   0                ;;USER DEFINED
  001202    000000              $TMP3:  .WORD   0                ;;USER DEFINED
  001204    000000              $TMP4:  .WORD   0                ;;USER DEFINED
  001206    000000              $TIMES: 0                        ;;MAX. NUMBER OF ITERATIONS
  001210    000000              $ESCAPE:0                        ;;ESCAPE ON ERROR ADDRESS
  001212       207      377  377 $BELL:  .ASCIZ  <207><377><377>  ;;CODE FOR BELL
  001216       077              $QUES:  .ASCII  /?/              ;;QUESTION MARK
  001217       015              $CRLF:  .ASCII  <15>             ;;CARRIAGE RETURN
  001220       012      000     $LF:    .ASCIZ  <12>             ;;LINE FEED
                                        ;;*****************************************************************
                                        .SBTTL  APT MAILBOX-ETABLE

                                        ;;*****************************************************************
                                        .EVEN
  001222                        $MAIL:                          ;;APT MAILBOX
  001222    000000              $MSGTY: .WORD   AMSGTY           ;;MESSAGE TYPE CODE
  001224    000000              $FATAL: .WORD   AFATAL           ;;FATAL ERROR NUMBER
  001226    000000              $TESTN: .WORD   ATESTN           ;;TEST NUMBER
```

```
001230  000000                  $PASS:   .WORD   APASS   ;;PASS COUNT
001232  000000                  $DEVCT:  .WORD   ADEVCT  ;;DEVICE COUNT
001234  000000                  $UNIT:   .WORD   AUNIT   ;;I/O UNIT NUMBER
001236  000000                  $MSGAD:  .WORD   AMSGAD  ;;MESSAGE ADDRESS
001240  000000                  $MSGLG:  .WORD   AMSGLG  ;;MESSAGE LENGTH
001242                          $ETABLE:                 ;;APT ENVIRONMENT TABLE
001242     000                  $ENV:    .BYTE   AENV    ;;ENVIRONMENT BYTE
001243     000                  $ENVM:   .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
001244  000000                  $SWREG:  .WORD   ASWREG  ;;APT SWITCH REGISTER
001246  000000                  $USWR:   .WORD   AUSWR   ;;USER SWITCHES
001250  000000                  $CPUOP:  .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                                ;*                       BITS 15-11=CPU TYPE
                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                              11/70=06,PDQ=07,Q=10
                                ;*                       BIT 10=REAL TIME CLOCK
                                ;*                       BIT  9=FLOATING POINT PROCESSOR
                                ;*                       BIT  8=MEMORY MANAGEMENT
001252     000                  $MAMS1:  .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001253     000                  $MTYP1:  .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                       MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                              900 NSEC CORE=001
                                ;*                              300 NSEC BIPOLAR=002
                                ;*                              500 NSEC MOS=003
001254  000000                  $MADR1:  .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                ;*                         MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
001256     000                  $MAMS2:  .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001257     000                  $MTYP2:  .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001260  000000                  $MADR2:  .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001262     000                  $MAMS3:  .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001263     000                  $MTYP3:  .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001264  000000                  $MADR3:  .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001266     000                  $MAMS4:  .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001267     000                  $MTYP4:  .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001270  000000                  $MADR4:  .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001272  120254                  $VECT1:  .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001274  000000                  $VECT2:  .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001276  176700                  $BASE:   .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001300  000000                  $DEVM:   .WORD   ADEVM   ;;DEVICE MAP
001302  000000                  $CDW1:   .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001304  000000                  $CDW2:   .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
001306  000000                  $DDW0:   .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001310  000000                  $DDW1:   .WORD   ADDW1   ;;DEVICE DESCRIPTOR.WORD#1
001312  000000                  $DDW2:   .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001314  000000                  $DDW3:   .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001316  000000                  $DDW4:   .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001320  000000                  $DDW5:   .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001322  000000                  $DDW6:   .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001324  000000                  $DDW7:   .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001326                          $ETEND:
                                .MEXIT
```

```
                                .SBTTL  USER DEFINED TAGS

001326  000000                  CTLFG:  .WORD   0       ;CONTAINS CONTROL-C FLAG
001330  000000                  CHGADR: .WORD   0       ;CHANGE RH/RM BUS ADDRESS = -1, NO CHANGE   0
001332  000000                  XXDP:   .WORD   0       ;THE LOW BYTE CONTAINS THE DRIVE NUMBER FROM WHICH
                                                        ;THE PROGRAM WAS LOADED. THE HIGH BYTE CONTAINS THE
                                                        ;'XXDP' DEVICE CODE FOR THE RM05/3/2.
001334      000                 _STRK:  .BYTE   0       ;LO BYTE = 0
001335      000                         .BYTE   0       ;HI BYTE, CONTAINS LAST TRACK ADDRESS OF UNIT
                                                        ;UNDER TEST. RM02/3 = 4., RM05 = 18.

                                ;THE REGISTER INPUT BUFFER IS USED FOR
                                ;STORING DRIVE STATUS

001336                          GETBUF:

                                ;REGISTER INPUT BUFFER
001336  000000                  RMCS1I: .WORD   0       ;CONTROL, STATUS REGISTER #1
001340  000000                  RMWCI:  .WORD   0       ;WORD COUNT REGISTER
001342  000000                  RMBAI:  .WORD   0       ;BUS ADDRESS REGISTER
001344  000000                  RMDAI:  .WORD   0       ;DISK ADDRESS REGISTER
001346  000000                  RMCS2I: .WORD   0       ;CONTROL, STATUS REGISTER #2
001350  000000                  RMDSI:  .WORD   0       ;DRIVE STATUS REGISTER
001352  000000                  RMER1I: .WORD   0       ;ERROR REGISTER #1
001354  000000                  RMASI:  .WORD   0       ;ATTENTION SUMMARY REGISTER
001356  000000                  RMLAI:  .WORD   0       ;LOOK AHEAD REGISTER
001360  000000                  RMDBI:  .WORD   0       ;DATA BUFFER
001362  000000                  RMMR1I: .WORD   0       ;MAINTENANCE REGISTER #1
001364  000000                  RMDTI:  .WORD   0       ;DRIVE TYPE REGISTER
001366  000000                  RMSNI:  .WORD   0       ;SERIAL NUMBER REGISTER
001370  000000                  RMOFI:  .WORD   0       ;OFFSET REGISTER
001372  000000                  RMDCI:  .WORD   0       ;DESIRED CYLINDER REGISTER
001374  000000                  RMHRI:  .WORD   0       ;HOLDING REGISTER
001376  000000                  RMMR2I: .WORD   0       ;MAINTENANCE REGISTER #2
001400  000000                  RMER2I: .WORD   0       ;ERROR REGISTER #2
001402  000000                  RMEC1I: .WORD   0       ;ECC POSITION REGISTER
001404  000000                  RMEC2I: .WORD   0       ;ECC PATTERN REGISTER
001406  000000                  RMBAEI: .WORD   0       ;BUS ADDRESS EXTENSION REGISTER
001410  000000                  RMCS3I: .WORD   0       ;CONTROL, STATUS REGISTER #3

                                ;THE REGISTER OUTPUT BUFFER IS USED FOR
                                ;ASSEMBLING DATA GOING TO REGISTER

001412                          PUTBUF:

                                ;REGISTER OUTPUT BUFFER
001412  000000                  RMCS1O: .WORD   0       ;CONTROL, STATUS REGISTER #1
001414  000000                  RMWCO:  .WORD   0       ;WORD COUNT REGISTER
001416  000000                  RMBAO:  .WORD   0       ;BUS ADDRESS REGISTER
001420  000000                  RMDAO:  .WORD   0       ;DISK ADDRESS REGISTER
001422  000000                  RMCS2O: .WORD   0       ;CONTROL, STATUS REGISTER #2
001424  000000                  RMDSO:  .WORD   0       ;DRIVE STATUS REGISTER
001426  000000                  RMER1O: .WORD   0       ;ERROR REGISTER #1
001430  000000                  RMASO:  .WORD   0       ;ATTENTION SUMMARY REGISTER
001432  000000                  RMLAO:  .WORD   0       ;LOOK AHEAD REGISTER
001434  000000                  RMDBO:  .WORD   0       ;DATA BUFFER
001436  000000                  RMMR1O: .WORD   0       ;MAINTENANCE REGISTER #1
```

```
001440  000000                  RMDTO:  .WORD   0       ;DRIVE TYPE REGISTER
001442  000000                  RMSNO:  .WORD   0       ;SERIAL NUMBER REGISTER
001444  000000                  RMOFO:  .WORD   0       ;OFFSET REGISTER
001446  000000                  RMDCO:  .WORD   0       ;DESIRED CYLINDER REGISTER
001450  000000                  RMHRO:  .WORD   0       ;HOLDING REGISTER
001452  000000                  RMMR2O: .WORD   0       ;MAINTENANCE REGISTER #2
001454  000000                  RMER2O: .WORD   0       ;ERROR REGISTER #2
001456  000000                  RMEC1O: .WORD   0       ;ECC POSITION REGISTER
001460  000000                  RMEC2O: .WORD   0       ;ECC PATTERN REGISTER
001462  000000                  RMBAEO: .WORD   0       ;BUS ADDRESS EXTENSION REGISTER
001464  000000                  RMCS3O: .WORD   0       ;CONTROL, STATUS REGISTER #3

                                ;EACH WORD OF THE TEST QUE CONTAINS THE DEVICE NUMBER IN
                                ;THE LOW BYTE AND THE ATTENTION BIT IN THE HIGH BYTE. THE
                                ;FIRST WORD CONTAINS THE ADDRESS OF THE DEVICE UNDER TEST
                                ;IN THE TABLE. A ZERO WORD IS A BLANK AND REPRESENTS THE
                                ;END OF THE QUE.

001466  000000                  TSTQUE: .WORD   0               ;CONTAINS DEVICE POINTER
001470                                  .BLKW   8.              ;TEST QUE FOR DEVICES UNDER TEST
001510  000000                          .WORD   0               ;TABLE TERMINATOR GOES HERE WHEN
                                                                ;ALL 8. DEVICES ARE UNDER TEST.

                                ;MEDIA ENABL  IS SET IF THE BAD SECTOR FILES HAVE BEEN RECOVERED
                                ;FOR THE UNIT UNDER TEST, OTHERWISE IT IS ZERO.
001512  000000                  MEDENB: .WORD   0               ;MEDIA ENABLE

                                ;LOCATIONS "ASNDC" AND "ASNDC" CONTAIN THE CYLINDER, TRACK AND SECTOR
                                ;ADDRESS ASSIGNED BY THE BAD SECTOR MODULE.
001514  000000                  ASNDC:  .WORD   0               ;ASSIGNED DESIRED CYLINDER
001516  000000                  ASNDA:  .WORD   0               ;ASSIGNED TRACK, AND SECTOR
001520  000000                  CLKADR: .WORD   0               ;UNIBUS ADDRESS OF KW11 CLOCK
001522  000000                  CLKVCT: .WORD   0               ;VECTOR ADDRESS OF KW11 CLOCK

                                ;THE GET INDEX TABLE CONTAINS A BYTE LIST OF REGISTERS WHICH
                                ;ARE READ BY THE GET SUBROUTINE.  THE LIST IS TERMINATED BY
                                ;A NEGATIVE BYTE.
001524                          GETINX: .BLKB   23.             ;GET INDEX TABLE

                                ;THE PUT INDEX TABLE ICONTAINS A BYTE LIST OF REGISTERS WHICH
                                ;ARE WRITTEN BY THE PUT SUBROUTINE.  THE LIST IS TERMINATED BY
                                ;A NEGATIVE BYTE.
001553                          PUTINX: .BLKB   23.             ;PUT INDEX TABLE

                                ;PUT TAGS HERE
```

```
                                .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


  001602                        $ERRTB:
  1
  2                             ;ERROR  1       WRONG UNIT SELECTED
  3
  001602  072404                        EMT1
  001604  076470                        EHT1
  001606  076614                        EDT1
  001610  076704                        EFT1

  4
  5                             ;ERROR  2       DEVICE WENT UNAVAILABLE
  6
  001612  072410                        EMT2
  001614  076470                        EHT1
  001616  076614                        EDT1
  001620  076704                        EFT1

  7
  8                             ;ERROR  3       DEVICE WENT NONEXISTENT
  9
  001622  072416                        EMT3
  001624  076470                        EHT1
  001626  076614                        EDT1
  001630  076704                        EFT1

 10
 11                             ;ERROR  4       CONTROLLER NOT READY
 12
  001632  072424                        EMT4
  001634  076470                        EHT1
  001636  076614                        EDT1
  001640  076704                        EFT1

 13
 14                             ;ERROR  5       DRIVE NOT READY AND GO NOT RESET
 15
  001642  072432                        EMT5
  001644  076470                        EHT1
  001646  076614                        EDT1
  001650  076704                        EFT1

 16
 17                             ;ERROR  6       UNEXPECTED VALUE FOR "ATA" STATUS
```

```
 18
 001652  072440                          EMT6
 001654  076470                          EHT1
 001656  076614                          EDT1
 001660  076704                          EFT1

 19
 20                             ;ERROR  7       BUS TIMEOUT TRYING TO READ OR WRITE REGISTER
 21
 001662  072446                          EMT7
 001664  000000                          0
 001666  000000                          0
 001670  000000                          0

 22
 23                             ;ERROR  10      DRIVE NOT READY BUT GO IS RESET
 24
 001672  072454                          EMT10
 001674  076470                          EHT1
 001676  076614                          EDT1
 001700  076704                          EFT1

 25
 26                             ;ERROR  11      GO NOT RESET BUT DRIVE IS READY
 27
 001702  072460                          EMT11
 001704  076470                          EHT1
 001706  076614                          EDT1
 001710  076704                          EFT1

 28
 29                             ;ERROR  12      INCORRECT FUNCTION CODE
 30
 001712  072464                          EMT12
 001714  076470                          EHT1
 001716  076614                          EDT1
 001720  076704                          EFT1

 31
 32                             ;ERROR  13      PARITY ERROR READING REMOTE REGISTERS
 33
 001722  072472                          EMT13
 001724  076470                          EHT1
 001726  076614                          EDT1
 001730  076704                          EFT1

 34
 35                             ;ERROR  14      TRANSFER ERROR IS INCORRECT
 36
 001732  072504                          EMT14
 001734  076470                          EHT1
 001736  076614                          EDT1
 001740  076704                          EFT1

 37
 38                             ;ERROR  15      INCORRECT WORD COUNT
 39
```

```
                001742  072512                          EMT15
                001744  076470                          EHT1
                001746  076614                          EDT1
                001750  076704                          EFT1

        40
        41                              ;ERROR  16      INCORRECT BUS ADDRESS
        42
                001752  072520                          EMT16
                001754  076470                          EHT1
                001756  076614                          EDT1
                001760  076704                          EFT1

        43
        44                              ;ERROR  17      INCORRECT LBT STATUS
        45
                001762  072530                          EMT17
                001764  076470                          EHT1
                001766  076614                          EDT1
                001770  076704                          EFT1

        46
        47                              ;ERROR  20      INCORRECT AOE
        48
                001772  072540                          EMT20
                001774  076470                          EHT1
                001776  076614                          EDT1
                002000  076704                          EFT1

        49
        50                              ;ERROR  21      INCORRECT DISK ADDRESS
        51
                002002  072550                          EMT21
                002004  076470                          EHT1
                002006  076614                          EDT1
                002010  076704                          EFT1

        52
        53                              ;ERROR  22      INCORRECT CYLINDER ADDRESS
        54
                002012  072560                          EMT22
                002014  076470                          EHT1
                002016  076614                          EDT1
                002020  076704                          EFT1

        55
        56                              ;ERROR  23      INCORRECT WLE STATUS
        57
                002022  072570                          EMT23
                002024  076470                          EHT1
                002026  076614                          EDT1
                002030  076704                          EFT1

        58
        59                              ;ERROR  24      INCORRECT UPE STATUS
        60
                002032  072600                          EMT24
```

```
002034  076470                          EHT1
002036  076614                          EDT1
002040  076704                          EFT1

61
62                              ;ERROR  25      INCORRECT WCF STATUS
63
002042  072610                          EMT25
002044  076470                          EHT1
002046  076614                          EDT1
002050  076704                          EFT1

64
65                              ;ERROR  26      INCORRECT WCE STATUS
66
002052  072620                          EMT26
002054  076470                          EHT1
002056  076614                          EDT1
002060  076704                          EFT1

67
68                              ;ERROR  27      INCORRECT MDPE STATUS
69
002062  072630                          EMT27
002064  076470                          EHT1
002066  076614                          EDT1
002070  076704                          EFT1

70
71                              ;ERROR  30      INCORRECT DCK STATUS
72
002072  072640                          EMT30
002074  076470                          EHT1
002076  076614                          EDT1
002100  076704                          EFT1

73
74                              ;ERROR  31      INCORRECT ECH STATUS
75
002102  072650                          EMT31
002104  076470                          EHT1
002106  076614                          EDT1
002110  076704                          EFT1

76
77                              ;ERROR  32      DLT SHOULD NOT BE SET
78
002112  072660                          EMT32
002114  076470                          EHT1
002116  076614                          EDT1
002120  076704                          EFT1

79
80                              ;ERROR  33      MXF SHOULD NOT BE SET
81
002122  072670                          EMT33
002124  076470                          EHT1
```

```
                002126  076614                  EDT1
                002130  076704                  EFT1

     82
     83                                 ;ERROR  34      DTE SHOULD NOT BE SET
     84
                002132  072700                  EMT34
                002134  076470                  EHT1
                002136  076614                  EDT1
                002140  076704                  EFT1

     85
     86                                 ;ERROR  35      INCORRECT HCRC STATUS
     87
                002142  072710                  EMT35
                002144  076470                  EHT1
                002146  076614                  EDT1
                002150  076704                  EFT1

     88
     89                                 ;ERROR  36      INCORRECT HCE STATUS
     90
                002152  072720                  EMT36
                002154  076470                  EHT1
                002156  076614                  EDT1
                002160  076704                  EFT1

     91
     92                                 ;ERROR  37      INCORRECT FER STATUS
     93
                002162  072730                  EMT37
                002164  076470                  EHT1
                002166  076614                  EDT1
                002170  076704                  EFT1

     94
     95                                 ;ERROR  40      DPE SHOULD NOT BE SET (NOT A DATA COMMAND)
     96
                002172  072740                  EMT40
                002174  076470                  EHT1
                002176  076614                  EDT1
                002200  076704                  EFT1

     97
     98                                 ;ERROR  41      LOST 'MOL' DURING PACK ACKNOWLEDGE
     99
                002202  072746                  EMT41
                002204  076470                  EHT1
                002206  076614                  EDT1
                002210  076704                  EFT1

    100
    101                                 ;ERROR  42      UNSAFE ERROR DURING PACK ACKNOWLEDGE
    102
                002212  072756                  EMT42
                002214  076470                  EHT1
                002216  076614                  EDT1
```

```
        002220  076704                          EFT1

103
104             ,                       ;ERROR  43      'OPI' ERROR DURING PACK ACKNOWLEDGE
105
        002222  072770                          EMT43
        002224  076470                          EHT1
        002226  076614                          EDT1
        002230  076704                          EFT1

106
107                                     ;ERROR  44      'RMR' ERROR DURING PACK ACKNOWLEDGE
108
        002232  073000                          EMT44
        002234  076470                          EHT1
        002236  076614                          EDT1
        002240  076704                          EFT1

109
110                                     ;ERROR  45      'ILR' ERROR DURING PACK ACKNOWLEDGE
111
        002242  073010                          EMT45
        002244  076470                          EHT1
        002246  076614                          EDT1
        002250  076704                          EFT1

112
113                                     ;ERROR  46      'ILF' ERROR DURING PACK ACKNOWLEDGE
114
        002252  073020                          EMT46
        002254  076470                          EHT1
        002256  076614                          EDT1
        002260  076704                          EFT1

115
116                                     ;ERROR  47      COMPOSITE ERROR STATUS IS INCORRECT
117
        002262  073030                          EMT47
        002264  076470                          EHT1
        002266  076614                          EDT1
        002270  076704                          EFT1

118
119                                     ;ERROR  50      PARITY ERROR WRITING REMOTE REGISTERS
120
        002272  073036                          EMT50
        002274  076470                          EHT1
        002276  076614                          EDT1
        002300  076704                          EFT1

121
122                                     ;ERROR  51      INCORRECT IAE STATUS DURING SEEK COMMAND
123
        002302  073046                          EMT51
        002304  076470                          EHT1
        002306  076614                          EDT1
        002310  076704                          EFT1
```

```
124
125                                     ;ERROR  52      OPI ERROR DURING SEEK - MEDIUM IS NOT ON LINE
126
    002312  073060                                    EMT52
    002314  076470                                    EHT1
    002316  076614                                    EDT1
    002320  076704                                    EFT1

127
128                                     ;ERROR  53      OPI ERROR DURING SEEK - MEDIUM IS ON LINE, ASSUME
129                                     ;               ON CYLINDER LATCH DIDN'T RESET
130
    002322  073076                                    EMT53
    002324  076470                                    EHT1
    002326  076614                                    EDT1
    002330  076704                                    EFT1

131
132                                     ;ERROR  54      SEEK INCOMPLETE ERROR DURING SEEK COMMAND
133
    002332  073114                                    EMT54
    002334  076470                                    EHT1
    002336  076614                                    EDT1
    002340  076704                                    EFT1

134
135                                     ;ERROR  55      DEVICE CHECK DURING SEEK COMMAND
136
    002342  073124                                    EMT55
    002344  076470                                    EHT1
    002346  076614                                    EDT1
    002350  076704                                    EFT1

137
138                                     ;ERROR  56      PIP IS STILL SET AFTER SEEK - SKI IS RESET
139
    002352  073136                                    EMT56
    002354  076470                                    EHT1
    002356  076614                                    EDT1
    002360  076704                                    EFT1

140
141                                     ;ERROR  57      ATA DID NOT SET DURING SEEK COMMAND
142
    002362  073154                                    EMT57
    002364  076470                                    EHT1
    002366  076614                                    EDT1
    002370  076704                                    EFT1

143
144                                     ;ERROR  60      IVC ERROR DURING SEEK COMMAND - LOST
145                                     ;               VOLUME VALID
146
    002372  073164                                    EMT60
    002374  076470                                    EHT1
    002376  076614                                    EDT1
```

```
        002400  076704                          EFT1

    147
    148                                 ;ERROR  61      ERRONEOUS IVC ERROR DURING SEEK COMMAND -
    149                                 ;               VOLUME VALID IS STIL SET
    150
        002402  073202                          EMT61
        002404  076470                          EHT1
        002406  076614                          EDT1
        002410  076704                          EFT1

    151
    152                                 ;ERROR  62      MOL IS ZERO, BUT OPI WAS NOT
    153                                 ;               REPORTED DURING SEEK COMMAND
    154
        002412  073222                          EMT62
        002414  076470                          EHT1
        002416  076614                          EDT1
        002420  076704                          EFT1

    155
    156                                 ;ERROR  63      UNUSED
    157
        002422  000000                          0
        002424  000000                          0
        002426  000000                          0
        002430  000000                          0

    158
    159                                 ;ERROR  64      DRIVE DID NOT DETECT ''IVC'' ERROR DURING SEEK
    160
        002432  073240                          EMT64
        002434  076470                          EHT1
        002436  076614                          EDT1
        002440  076704                          EFT1

    161
    162                                 ;ERROR  65      DRIVE EXECUTED A SEEK WITH ERROR SET
    163
        002442  073260                          EMT65
        002444  076470                          EHT1
        002446  076614                          EDT1
        002450  076704                          EFT1

    164
    165                                 ;ERROR  66      UNEXPECTED ERROR SET IN RMER1
    166
        002452  073300                          EMT66
        002454  076470                          EHT1
        002456  076614                          EDT1
        002460  076704                          EFT1

    167
    168                                 ;ERROR  67      UNEXPECTED ERROR SET IN RMER2
    169
        002462  073312                          EMT67
        002464  076470                          EHT1
```

```
002466  076614                          EDT1
002470  076704                          EFT1

170
171                             ;ERROR  70      ERRONEOUS ''IAE'' ERROR DURING RECALIBRATE
172
002472  073324                          EMT70
002474  076470                          EHT1
002476  076614                          EDT1
002500  076704                          EFT1

173
174                             ;ERROR  71      ''ILF'' ERROR DURING RECALIBRATE
175
002502  073334                          EMT71
002504  076470                          EHT1
002506  076614                          EDT1
002510  076704                          EFT1

176
177                             ;ERROR  72      'OPI'' ERROR DURING RECALIBRATE DUE TO 'MOL'' = 0
178
002512  073344                          EMT72
002514  076470                          EHT1
002516  076614                          EDT1
002520  076704                          EFT1

179
180                             ;ERROR  73      'OPI'' ERROR DURING RECALIBRATE BECAUSE ON
181                             ;               CYLINDER DIDNT DROP
182
002522  073362                          EMT73
002524  076470                          EHT1
002526  076614                          EDT1
002530  076704                          EFT1

183
184                             ;ERROR  74      ''IVC'' ERROR DURING RECALIBRATE - 'VV'' = 0
185
002532  073400                          EMT74
002534  076470                          EHT1
002536  076614                          EDT1
002540  076704                          EFT1

186
187                             ;ERROR  75      ERRONEOUS ''IVC'' ERROR DURING RECALIBRATE - 'VV''  1
188
002542  073410                          EMT75
002544  076470                          EHT1
002546  076614                          EDT1
002550  076704                          EFT1

189
190                             ;ERROR  76      ''SKI'' ERROR DURING RECALIBRATE
191
002552  073430                          EMT76
002554  076470                          EHT1
```

```
002556  076614                          EDT1
002560  076704                          EFT1

192
193                             ;ERROR  77      "DVC" OCCURRED DURING RECALIBRATE
194
002562  073440                          EMT77
002564  076470                          EHT1
002566  076614                          EDT1
002570  076704                          EFT1

195
196                             ;ERROR  100     LOST "MOL" DURING RECALIBRATE - "OPI"   0
197
002572  073452                          EMT100
002574  076470                          EHT1
002576  076614                          EDT1
002600  076704                          EFT1

198
199                             ;ERROR  101     LOST "VV" DURING RECALIBRATE - "IVC" = 0
200
002602  073470                          EMT101
002604  076470                          EHT1
002606  076614                          EDT1
002610  076704                          EFT1

201
202                             ;ERROR  102     "ATA" DID NOT SET DURING RECALIBRATE
203
002612  073506                          EMT102
002614  076470                          EHT1
002616  076614                          EDT1
002620  076704                          EFT1

204
205                             ;ERROR  103     "OM" DID NOT RESET DURING RECALIBRATE
206
002622  073516                          EMT103
002624  076470                          EHT1
002626  076614                          EDT1
002630  076704                          EFT1

207
208                             ;ERROR  104     "PIP" IS STIL SET AFTER RECALIBRATE
209
002632  073530                          EMT104
002634  076470                          EHT1
002636  076614                          EDT1
002640  076704                          EFT1

210
211                             ;ERROR  105     UNEXPECTED "ILR" ERROR DURING RECALIBRATE
212
002642  073546                          EMT105
002644  076470                          EHT1
002646  076614                          EDT1
```

```
        002650  076704                          EFT1

    213
    214                             ;ERROR  106     UNEXPECTED 'RMR' ERROR DURING RECALIBRATE
    215
        002652  073556                          EMT106
        002654  076470                          EHT1
        002656  076614                          EDT1
        002660  076704                          EFT1

    216
    217                             ;ERROR  107     'UNS' ERROR DURING RECALIBRATE - AC POWER IS LOW
    218
        002662  073566                          EMT107
        002664  076470                          EHT1
        002666  076614                          EDT1
        002670  076704                          EFT1

    219
    220                             ;ERROR  110     CANNOT ACCESS MASSBUS CONTROLLER VIA UNIBUS
    221
        002672  073606                          EMT110
        002674  076512                          EHT110
        002676  076632                          EDT110
        002700  076722                          EFT110

    222
    223                             ;ERROR  111     NONEXISTENT DEVICE
    224
        002702  073620                          EMT111
        002704  076516                          EHT111
        002706  076634                          EDT111
        002710  076724                          EFT111

    225
    226                             ;ERROR  112     DEVICE NOT AVAILABLE
    227
        002712  073626                          EMT112
        002714  076516                          EHT111
        002716  076634                          EDT111
        002720  076724                          EFT111

    228
    229                             ;ERROR  113     BUS TIMEOUT-NED STATUS FAILURE
    230
        002722  073634                          EMT113
        002724  000000                          0
        002726  000000                          0
        002730  000000                          0

    231
    232                             ;ERROR  114     DEVICE NOT AN RM05/3/2
    233
        002732  073650                          EMT114
        002734  076522                          EHT114
        002736  076636                          EDT114
        002740  076726                          EFT114
```

```
234
235                             ;ERROR  115     RMCS1 NOT INITIALIZED BY UNIBUS
236
     002742  073656                             EMT115
     002744  076470                             EHT1
     002746  076614                             EDT1
     002750  076704                             EFT1

237
238                             ;ERROR  116     RMBA NOT INITIALIZED BY UNIBUS
239
     002752  073666                             EMT116
     002754  076470                             EHT1
     002756  076614                             EDT1
     002760  076704                             EFT1

240
241                             ;ERROR  117     RMCS2 NOT INITIALIZED BY UNIBUS
242
     002762  073676                             EMT117
     002764  076470                             EHT1
     002766  076614                             EDT1
     002770  076704                             EFT1

243
244                             ;ERROR  120     RMER1 NOT INITIALIZED BY UNIBUS
245
     002772  073706                             EMT120
     002774  076470                             EHT1
     002776  076614                             EDT1
     003000  076704                             EFT1

246
247                             ;ERROR  121     RMAS NOT INITIALIZED BY UNIBUS
248
     003002  073716                             EMT121
     003004  076470                             EHT1
     003006  076614                             EDT1
     003010  076704                             EFT1

249
250                             ;ERROR  122     RMMR1 NOT INITIALIZED BY UNIBUS
251
     003012  073726                             EMT122
     003014  076470                             EHT1
     003016  076614                             EDT1
     003020  076704                             EFT1

252
253                             ;ERROR  123     RMDS NOT INITIALIZED BY UNIBUS
254
     003022  073736                             EMT123
     003024  076470                             EHT1
     003026  076614                             EDT1
     003030  076704                             EFT1
```

```
255
256                                     ;ERROR  124     RMEC2 NOT INITIALIZED BY UNIBUS
257
     003032  073746                               EMT124
     003034  076470                               EHT1
     003036  076614                               EDT1
     003040  076704                               EFT1

258
259                                     ;ERROR  125     RMMR2 NOT INITIALIZED BY UNIBUS
260
     003042  073756                               EMT125
     003044  076470                               EHT1
     003046  076614                               EDT1
     003050  076704                               EFT1

261
262                                     ;ERROR  126     RMCS1 NOT CLEARED BY CONTROLLER CLEAR
263
     003052  073766                               EMT126
     003054  076470                               EHT1
     003056  076614                               EDT1
     003060  076704                               EFT1

264
265                                     ;ERROR  127     RMBA NOT CLEARED BY CONTROLLER CLEAR
266
     003062  074000                               EMT127
     003064  076470                               EHT1
     003066  076614                               EDT1
     003070  076704                               EFT1

267
268                                     ;ERROR  130     RMCS2 NOT CLEARED BY CONTROLLER CLEAR
269
     003072  074012                               EMT130
     003074  076470                               EHT1
     003076  076614                               EDT1
     003100  076704                               EFT1

270
271                                     ;ERROR  131     RMER1 NOT CLEARED BY CONTROLLER CLEAR
272
     003102  074024                               EMT131
     003104  076470                               EHT1
     003106  076614                               EDT1
     003110  076704                               EFT1

273
274                                     ;ERROR  132     RMAS NOT CLEARED BY CONTROLLER CLEAR
275
     003112  074036                               EMT132
     003114  076470                               EHT1
     003116  076614                               EDT1
     003120  076704                               EFT1

276
```

```
277                                     ;ERROR  133     RMMR1 NOT CLEARED BY CONTROLLER CLEAR
278
     003122  074050                             EMT133
     003124  076470                             EHT1
     003126  076614                             EDT1
     003130  076704                             EFT1

279
280                                     ;ERROR  134     RMDS NOT CLEARED BY CONTROLLER CLEAR
281
     003132  074062                             EMT134
     003134  076470                             EHT1
     003136  076614                             EDT1
     003140  076704                             EFT1

282
283                                     ;ERROR  135     RMEC2 NOT CLEARED BY CONTROLLER CLEAR
284
     003142  074074                             EMT135
     003144  076470                             EHT1
     003146  076614                             EDT1
     003150  076704                             EFT1

285
286                                     ;ERROR  136     RMMR2 NOT CLEARED BY CONTROLLER CLEAR
287
     003152  074106                             EMT136
     003154  076470                             EHT1
     003156  076614                             EDT1
     003160  076704                             EFT1

288
289                                     ;ERROR  137     RMCS1 NOT CLEARED BY ERROR CLEAR
290
     003162  074120                             EMT137
     003164  076470                             EHT1
     003166  076614                             EDT1
     003170  076704                             EFT1

291
292                                     ;ERROR  140     RMCS2 NOT CLEARED BY ERROR CLEAR
293
     003172  074130                             EMT140
     003174  076470                             EHT1
     003176  076614                             EDT1
     003200  076704                             EFT1

294
295                                     ;ERROR  141     RMCS1 NOT CLEARED BY DRIVE CLEAR
296
     003202  074140                             EMT141
     003204  076470                             EHT1
     003206  076614                             EDT1
     003210  076704                             EFT1

297
298                                     ;ERROR  142     RMDS NOT CLEARED BY DRIVE CLEAR
```

```
299
    003212  074150                          EMT142
    003214  076470                          EHT1
    003216  076614                          EDT1
    003220  076704                          EFT1

300
301                             ;ERROR  143     RMER1 NOT CLEARED BY DRIVE CLEAR
302
    003222  074160                          EMT143
    003224  076470                          EHT1
    003226  076614                          EDT1
    003230  076704                          EFT1

303
304                             ;ERROR  144     RMAS NOT CLEARED BY DRIVE CLEAR
305
    003232  074170                          EMT144
    003234  076470                          EHT1
    003236  076614                          EDT1
    003240  076704                          EFT1

306
307                             ;ERROR  145     RMMR1  NOT CLEARED BY DRIVE CLEAR
308
    003242  074200                          EMT145
    003244  076470                          EHT1
    003246  076614                          EDT1
    003250  076704                          EFT1

309
310                             ;ERROR  146     RMMR2 NOT CLEARED BY DRIVE CLEAR
311
    003252  074210                          EMT146
    003254  076470                          EHT1
    003256  076614                          EDT1
    003260  076704                          EFT1

312
313                             ;ERROR  147     RMER2 NOT CLEARED BY DRIVE CLEAR
314
    003262  074220                          EMT147
    003264  076470                          EHT1
    003266  076614                          EDT1
    003270  076704                          EFT1

315
316                             ;ERROR  150     RMEC2 NOT CLEARED BY DRIVE CLEAR
317
    003272  074230                          EMT150
    003274  076470                          EHT1
    003276  076614                          EDT1
    003300  076704                          EFT1

318
319                             ;ERROR  151     MEDIUM NOT ON LINE
320
```

```
003302  074240                          EMT151
003304  076470                          EHT1
003306  076614                          EDT1
003310  076704                          EFT1

322
322                             ;ERROR  152     DRIVE FAULT
323
003312  074252                          EMT152
003314  076470                          EHT1
003316  076614                          EDT1
003320  076704                          EFT1

324
325                             ;ERROR  153     UNSAFE SHOULD BE SET BECAUSE DVC IS SET
326
003322  074264                          EMT153
003324  076470                          EHT1
003326  076614                          EDT1
003330  076704                          EFT1

327
328                             ;ERROR  154     UNSAFE SHOULD NOT BE SET, AC IS LOW
329
003332  074302                          EMT154
003334  076470                          EHT1
003336  076614                          EDT1
003340  076704                          EFT1

330
331                             ;ERROR  155     VOLUME VALID NOT SET BY PACK ACK
332
003342  074320                          EMT155
003344  076470                          EHT1
003346  076614                          EDT1
003350  076704                          EFT1

333
334                             ;ERROR  156     OFFSET MODE NOT SET BY OFFSET COMMAND
335
003352  074332                          EMT156
003354  076470                          EHT1
003356  076614                          EDT1
003360  076704                          EFT1

336
337                             ;ERROR  157     OFFSET MODE NOT RESET BY RTC COMMAND
338
003362  074344                          EMT157
003364  076470                          EHT1
003366  076614                          EDT1
003370  076704                          EFT1

339
340                             ;ERROR  160     RMOF NOT RESET BY RIP COMMAND
341
003372  074356                          EMT160
```

```
003374  076470                          EHT1
003376  076614                          EDT1
003400  076704                          EFT1

342
343                             ;ERROR  161     RMDA NOT RESET BY RIP COMMAND
344
003402  074366                          EMT^161
003404  076470                          EHT1
003406  076614                          EDT1
003410  076704                          EFT1

345
346                             ;ERROR  162     RMDC NOT RESET BY RIP COMMAND
347
003412  074400                          EMT^62
003414  076470                          EHT1
003416  076614                          EDT1
003420  076704                          EFT1

348
349                             ;ERROR 163      DATA WAS ECC CORRECTED BUT DOES NOT COMPARE WITH
350                             ;               WRITE BUFFER
351
003422  076264                          EMT336
003424  076552                          EHT336
003426  076650                          EDT336
003430  076740                          EFT336

352
353                             ;ERROR  164     OPI SHOULD NOT BE SET
354
003432  074422                          EMT164
003434  076470                          EHT1
003436  076614                          EDT1
003440  076704                          EFT1

355
356                             ;ERROR  165     IVC SHOULD NOT BE SET
357
003442  074430                          EMT165
003444  076470                          EHT1
003446  076614                          EDT1
003450  076704                          EFT1

358
359                             ;ERROR  166     IAE SHOULD NOT BE SET
360
003452  074436                          EMT166
003454  076470                          EHT1
003456  076614                          EDT1
003460  076704                          EFT1

361
362                             ;ERROR  167     NEM SHOULD NOT BE SET
363
003462  074444                          EMT167
```

```
003464  076470                          EHT1
003466  076614                          EDT1
003470  076704                          EFT1

364
365                             ;ERROR  170     INCORRECT 'MOL'' STATUS DURING DIAGNOSTIC MODE
366
003472  074452                          EMT170
003474  076470                          EHT1
003476  076614                          EDT1
003500  076704                          EFT1

367
368                             ;ERROR  171     ''ATA'' NOT SET DURING RETURN TO CENTERLINE
369
003502  074464                          EMT171
003504  076470                          EHT1
003506  076614                          EDT1
003510  076704                          EFT1

370
371                             ;ERROR  172     ''ATA'' NOT SET BY OFFSET COMMAND
372
003512  074474                          EMT172
003514  076470                          EHT1
003516  076614                          EDT1
003520  076704                          EFT1

373
374                             ;ERROR  173     RMER2 NOT INITIALIZED BY UNIBUS INIT
375
003522  074504                          EMT173
003524  076470                          EHT1
003526  076614                          EDT1
003530  076704                          EFT1

376
377                             ;ERROR  174     RMER2 NOT INITIALIZED BY CONTROLLER CLEAR
378
003532  074514                          EMT174
003534  076470                          EHT1
003536  076614                          EDT1
003540  076704                          EFT1

379
380                             ;ERROR  175     SELECTED DEVICE IS IN WRITE PROTECT
381
003542  074526                          EMT175
003544  076470                          EHT1
003546  076614                          EDT1
003550  076704                          EFT1

382
383                             ;ERROR  176     CANNOT SET DIAGNOSTIC MODE
384
003552  074534                          EMT176
003554  076470                          EHT1
```

```
003556  076614                          EDT1
003560  076704                          EFT1

385
386                             ;ERROR  177     --RESERVED FOR POWER MONITOR BIT FAILURE--
387
003562  000000                          0
003564  000000                          0
003566  000000                          0
003570  000000                          0

388
389                             ;ERROR  200     INCORRECT 'PIP' STATUS DURING DIAGNOSTIC MODE
390
003572  074544                          EMT200
003574  076470                          EHT1
003576  076614                          EDT1
003600  076704                          EFT1

391
392                             ;ERROR  201     INCORRECT 'WRL' STATUS DURING DIAGNOSTIC MODE
393
003602  074556                          EMT201
003604  076470                          EHT1
003606  076614                          EDT1
003610  076704                          EFT1

394
395                             ;ERROR  202     INCORRECT 'SKI' STATUS DURING DIAGNOSTIC MODE
396
003612  074570                          EMT202
003614  076470                          EHT1
003616  076614                          EDT1
003620  076704                          EFT1

397
398                             ;ERROR  203     INCORRECT 'DVC' STATUS DURING DIAGNOSTIC MODE
399
003622  074602                          EMT203
003624  076470                          EHT1
003626  076614                          EDT1
003630  076704                          EFT1

400
401                             ;ERROR  204     'VV' WAS NOT RESET BY MAINTENANCE UNIT READY
402
003632  074614                          EMT204
003634  076470                          EHT1
003636  076614                          EDT1
003640  076704                          EFT1

403
404                             ;ERROR  205     SELECTED DEVICE HAS A PERSISTENT 'SKI' ERROR
405
003642  074632                          EMT205
003644  076470                          EHT1
003646  076614                          EDT1
```

```
                003650  076704                          EFT1

406
407                                     ;ERROR  206     'LBC'' DID NOT SET DURING DIAGNOSTIC MODE
408
                003652  074642                          EMT206
                003654  076470                          EHT1
                003656  076614                          EDT1
                003660  076704                          EFT1

409
410                                     ;ERROR  207     UNEXPECTED LOSS OF 'MOL'' - MEDIUM IS OFF LINE
411
                003662  074652                          EMT207
                003664  076470                          EHT1
                003666  076614                          EDT1
                003670  076704                          EFT1

412
413                                     ;ERROR  210     UNEXPECTED LOSS OF VOLUME VALID - 'VV'' = 0
414
                003672  074664                          EMT210
                003674  076470                          EHT1
                003676  076614                          EDT1
                003700  076704                          EFT1

415
416                                     ;ERROR  211     UNEXPECTED MECHANICAL MOTION - 'PIP'' - 1
417
                003702  074672                          EMT211
                003704  076470                          EHT1
                003706  076614                          EDT1
                003710  076704                          EFT1

418
419                                     ;ERROR  212     UNEXPECTED DEVICE FAULT - 'DVC'' = 1
420
                003712  074706                          EMT212
                003714  076470                          EHT1
                003716  076614                          EDT1
                003720  076704                          EFT1

421
422                                     ;ERROR  213     UNEXPECTED SEEK INCOMPLETE ERROR - ''SKI'' = 1
423
                003722  074722                          EMT213
                003724  076470                          EHT1
                003726  076614                          EDT1
                003730  076704                          EFT1

424
425                                     ;ERROR  214     DRIVE EXECUTED A RECALIBRATE WITH ERROR SET
426
                003732  074732                          EMT214
                003734  076502                          EHT2
                003736  076624                          EDT2
                003740  076714                          EFT2
```

```
427
428                                     ;ERROR  215     DRIVE DID NOT DETECT ''IVC'' ERROR DURING RECALIBRATE
429
    003742  074752                              EMT215
    003744  076502                              EHT2
    003746  076624                              EDT2
    003750  076714                              EFT2

430
431                                     ;ERROR  216     INCORRECT ''IVC'' STATUS
432
    003752  074764                              EMT216
    003754  076470                              EHT1
    003756  076614                              EDT1
    003760  076704                              EFT1

433
434                                     ;ERROR  217     INCORRECT ''IAE'' STATUS
435
    003762  074774                              EMT217
    003764  076470                              EHT1
    003766  076614                              EDT1
    003770  076704                              EFT1

436
437                                     ;ERROR 220      INCORRECT 'WLE'' STATUS
438
    003772  075004                              EMT220
    003774  076470                              EHT1
    003776  076614                              EDT1
    004000  076704                              EFT1

439
440                                     ;ERROR  221     INCORRECT 'OPI'' STATUS
441
    004002  075014                              EMT221
    004004  076470                              EHT1
    004006  076614                              EDT1
    004010  076704                              EFT1

442
443                                     ;ERROR  222     RM DID NOT DETECT RMR ERROR
444
    004012  075024                              EMT222
    004014  076470                              EHT1
    004016  076614                              EDT1
    004020  076704                              EFT1

445
446                                     ;ERROR  223     RM DID NOT DETECT PARITY ERROR ON MASSBUS CONTROL BUS
447
    004022  075034                              EMT223
    004024  076526                              EHT223
    004026  076640                              EDT223
    004030  076730                              EFT223
```

```
448
449                                     ;ERROR  224     UNUSED
450
        004032  000000                            0
        004034  000000                            0
        004036  000000                            0
        004040  000000                            0

451
452                                     ;ERROR  225     UNUSED
453
        004042  000000                            0
        004044  000000                            0
        004046  000000                            0
        004050  000000                            0

454
455                                     ;ERROR  226     UNUSED
456
        004052  000000                            0
        004054  000000                            0
        004056  000000                            0
        004060  000000                            0

457
458                                     ;ERROR  227     UNUSED
459
        004062  000000                            0
        004064  000000                            0
        004066  000000                            0
        004070  000000                            0

460
461                                     ;ERROR  230     UNUSED
462
        004072  000000                            0
        004074  000000                            0
        004076  000000                            0
        004100  000000                            0

463
464                                     ;ERROR  231     UNUSED
465
        004102  000000                            0
        004104  000000                            0
        004106  000000                            0
        004110  000000                            0

466
467                                     ;ERROR  232     UNUSED
468
        004112  000000                            0
        004114  000000                            0
        004116  000000                            0
        004120  000000                            0

469
```

```
470                                     ;ERROR  233     UNUSED
471
        004122  000000                          0
        004124  000000                          0
        004126  000000                          0
        004130  000000                          0

472
473                                     ;ERROR  234     UNUSED
474
        004132  000000                          0
        004134  000000                          0
        004136  000000                          0
        004140  000000                          0

475
476                                     ;ERROR  235     UNUSED
477
        004142  000000                          0
        004144  000000                          0
        004146  000000                          0
        004150  000000                          0

478
479                                     ;ERROR  236     UNUSED
480
        004152  000000                          0
        004154  000000                          0
        004156  000000                          0
        004160  000000                          0

481
482                                     ;ERROR  237     UNUSED
483
        004162  000000                          0
        004164  000000                          0
        004166  000000                          0
        004170  000000                          0

484
485                                     ;ERROR  240     UNUSED
486
        004172  000000                          0
        004174  000000                          0
        004176  000000                          0
        004200  000000                          0

487
488                                     ;ERROR  241     UNUSED
489
        004202  000000                          0
        004204  000000                          0
        004206  000000                          0
        004210  000000                          0

490
491                                     ;ERROR  242     UNUSED
```

```
492
     004212  000000                          0
     004214  000000                          0
     004216  000000                          0
     004220  000000                          0

493
494                                 ;ERROR  243     UNUSED
495
     004222  000000                          0
     004224  000000                          0
     004226  000000                          0
     004230  000000                          0

496
497                                 ;ERROR  244     UNUSED
498
     004232  000000                          0
     004234  000000                          0
     004236  000000                          0
     004240  000000                          0

499
500                                 ;ERROR  245     UNUSED
501
     004242  000000                          0
     004244  000000                          0
     004246  000000                          0
     004250  000000                          0

502
503                                 ;ERROR  246     "ATA" NOT RESET BY GO WHEN "ERR" - 0
504
     004252  075110                          EMT246
     004254  076470                          EHT1
     004256  076614                          EDT1
     004260  076704                          EFT1

505
506                                 ;ERROR  247     "ATA" NOT RESET BY WRITING RMAS
507
     004262  075120                          EMT247
     004264  076470                          EHT1
     004266  076614                          EDT1
     004270  076704                          EFT1

508
509                                 ;ERROR  250     "ATA" WAS RESET BY GO WHEN "ERR" = 1
510
     004272  075132                          EMT250
     004274  076470                          EHT1
     004276  076614                          EDT1
     004300  076704                          EFT1

511
512                                 ;ERROR  251     PROGRAM INTERRUPT WAS NOT GENERATED
513
```

```
004302  075146                          EMT251
004304  076502                          EHT2
004306  076624                          EDT2
004310  076714                          EFT2

514
515                             ;ERROR  252     PROGRAM INTERRUPT SHOULD NOT HAVE BEEN GENERATED
516
004312  075154                          EMT252
004314  076502                          EHT2
004316  076624                          EDT2
004320  076714                          EFT2

517
518                             ;ERROR  253     OFFSET MODE WAS NOT RESET BY WRITING RMDC
519
004322  075162                          EMT253
004324  076470                          EHT1
004326  076614                          EDT1
004330  076704                          EFT1

520
521                             ;ERROR  254     INCORRECT ''ILF'' STATUS
522
004332  075200                          EMT254
004334  076470                          EHT1
004336  076614                          EDT1
004340  076704                          EFT1

523
524                             ;ERROR  255     INCORRECT ''ATA'' STATUS
525
004342  075210                          EMT255
004344  076470                          EHT1
004346  076614                          EDT1
004350  076704                          EFT1

526
527                             ;ERROR  256     INCORRECT ''ILR'' STATUS
528
004352  075220                          EMT256
004354  076540                          EHT256
004356  076640                          EDT223
004360  076730                          EFT223

529
530                             ;ERROR  257     INVALID IAE STATUS DURING SEARCH COMMAND
531
004362  075230                          EMT257
004364  076470                          EHT1
004366  076614                          EDT1
004370  076704                          EFT1

532
533                             ;ERROR  260     ''IVC'' WAS NOT DETECTED DURING SEARCH COMMAND
534
004372  075242                          EMT260
```

```
004374  076470                          EHT1
004376  076614                          EDT1
004400  076704                          EFT1

535
536                             ;ERROR  261     DRIVE EXECUTED SEARCH WITH ERROR SET
537
004402  075254                          EMT261
004404  076470                          EHT1
004406  076614                          EDT1
004410  076704                          EFT1

538
539                             ;ERROR  262     "LBC" ERROR NOT SET DURING DIAGNOSTIC MODE
540
004412  075274                          EMT262
004414  076470                          EHT1
004416  076614                          EDT1
004420  076704                          EFT1

541
542                             ;ERROR  263     "SKI" ERROR DURING SEARCH COMMAND
543
004422  075304                          EMT263
004424  076470                          EHT1
004426  076614                          EDT1
004430  076704                          EFT1

544
545                             ;ERROR  264     "IVC" ERROR DURING SEARCH - LOST VOLUME VALID
546
004432  075314                          EMT264
004434  076470                          EHT1
004436  076614                          EDT1
004440  076704                          EFT1

547
548                             ;ERROR  265     ERRONEOUS IVC ERROR DURING SEARCH - VOLUME IS VALID
549
004442  075334                          EMT265
004444  076470                          EHT1
004446  076614                          EDT1
004450  076704                          EFT1

550
551                             ;ERROR  266     DEVICE FAULT (DVC) DURING SEARCH
552
004452  075354                          EMT266
004454  076470                          EHT1
004456  076614                          EDT1
004460  076704                          EFT1

553
554                             ;ERROR  267     SKI SHOULD HAVE BEEN SET BECAUSE CYLINDER
555                             ;               ADDRESS IS TOO LARGE
556
004462  075366                          EMT267
```

```
        004464  076470                          EHT1
        004466  076614                          EDT1
        004470  076704                          EFT1

    557
    558                             ;ERROR  270     OPI ERROR DURING SEARCH BECAUSE MOL = 0
    559
        004472  075404                          EMT270
        004474  076470                          EHT1
        004476  076614                          EDT1
        004500  076704                          EFT1

    560
    561                             ;ERROR  271     OPI ERROR DURING SEARCH BECAUSE ON CYLINDER
    562                             ;               DIDN'T DROP
    563
        004502  075420                          EMT271
        004504  076470                          EHT1
        004506  076614                          EDT1
        004510  076704                          EFT1

    564
    565                             ;ERROR  272     LOST MOL DURING SEARCH, OPI IS NOT SET
    566
        004512  075436                          EMT272
        004514  076470                          EHT1
        004516  076614                          EDT1
        004520  076704                          EFT1

    567
    568                             ;ERROR  273     PIP STIL SET AFTER SEARCH
    569
        004522  075454                          EMT273
        004524  076470                          EHT1
        004526  076614                          EDT1
        004530  076704                          EFT1

    570
    571                             ;ERROR  274     PARITY ERROR OCCURRED WHILE WRITING REMOTE
    572                             ;               REGISTERS BUT MXF DID NOT SET
    573
        004532  075472                          EMT274
        004534  076470                          EHT1
        004536  076614                          EDT1
        004540  076704                          EFT1

    574
    575                             ;ERROR  275     MXF ERROR - COMPOSITE ERROR OCCURRED BEFORE DATA
    576                             ;               COMMAND STARTED
    577
        004542  075510                          EMT275
        004544  076470                          EHT1
        004546  076614                          EDT1
        004550  076704                          EFT1

    578
    579                             ;ERROR  276     ''OPI'' ERROR DURING DATA TRANSFER BECAUSE 'MOL'' WAS
```

```
                                ;                ZERO
                                
004552  075522                          EMT276
004554  076470                          EHT1
004556  076614                          EDT1
004560  076704                          EFT1

                                ;ERROR  277      "OPI" ERROR DURING DATA TRANSFER BECAUSE 1) ON
                                ;                CYLINDER DIDN'T DROP OR 2) SEARCH TIMED OUT OR
                                ;                3) RUN TIMED OUT

004562  075536                          EMT277
004564  076470                          EHT1
004566  076614                          EDT1
004570  076704                          EFT1

                                ;ERROR  300      "IVC" ERROR DURING DATA TRANSFER BECAUSE VOLUME
                                ;                WAS NOT VALID

004572  075554                          EMT300
004574  076470                          EHT1
004576  076614                          EDT1
004600  076704                          EFT1

                                ;ERROR  301      ERRONEOUS "IVC" ERROR DURING DATA TRANSFER - VOLUME
                                ;                IS VALID

004602  075574                          EMT301
004604  076470                          EHT1
004606  076614                          EDT1
004610  076704                          EFT1

                                ;ERROR  302      "ILR" ERROR DURING DATA TRANSFER

004612  075616                          EMT302
004614  076470                          EHT1
004616  076614                          EDT1
004620  076704                          EFT1

                                ;ERROR  303      "ILF" ERROR DURING DATA TRANSFER

004622  075630                          EMT303
004624  076470                          EHT1
004626  076614                          EDT1
004630  076704                          EFT1

                                ;ERROR  304      "RMR" ERROR DURING DATA TRANSFER

004632  075642                          EMT304
004634  076470                          EHT1
004636  076614                          EDT1
```

```
        004640  076704                          EFT1

604
605                                     ;ERROR  305     INCORRECT ''IAE'' STATUS DURING DATA TRANSFER
606
        004642  075654                          EMT305
        004644  076470                          EHT1
        004646  076614                          EDT1
        004650  076704                          EFT1

607
608                                     ;ERROR  306     ''SKI'' ERROR DURING DATA TRANSFER
609
        004652  075666                          EMT306
        004654  076470                          EHT1
        004656  076614                          EDT1
        004660  076704                          EFT1

610
611                                     ;ERROR  307     DRIVE DID NOT DETECT SKI ERROR DUE TO CYLINDER
612
        004662  075676                          EMT307
        004664  076470                          EHT1
        004666  076614                          EDT1
        004670  076704                          EFT1

613
614                                     ;ERROR  310     DEVICE FAULT DURING DATA TRANSFER
615
        004672  075716                          EMT310
        004674  076470                          EHT1
        004676  076614                          EDT1
        004700  076704                          EFT1

616
617                                     ;ERROR  311     LOSS OF BIT CLOCK DURING DATA TRANSFER
618
        004702  075730                          EMT311
        004704  076470                          EHT1
        004706  076614                          EDT1
        004710  076704                          EFT1

619
620                                     ;ERROR  312     LOSS OF SYSTEM CLOCK DURING DATA TRANSFER
621
        004712  075742                          EMT312
        004714  076470                          EHT1
        004716  076614                          EDT1
        004720  076704                          EFT1

622
623                                     ;ERROR  313     UNSAFE ERROR DURING DATA TRANSFER (DVC - 0)
624
        004722  075754                          EMT313
        004724  076470                          EHT1
        004726  076614                          EDT1
        004730  076704                          EFT1
```

```
                                ;ERROR  314     DRIVE TIMING ERROR DURING DATA TRANSFER
004732  075774                          EMT314
004734  076470                          EHT1
004736  076614                          EDT1
004740  076704                          EFT1

                                ;ERROR  315     WRITE LOCK ERROR
004742  076006                          EMT315
004744  076470                          EHT1
004746  076614                          EDT1
004750  076704                          EFT1

                                ;ERROR  316     ERRONEOUS WRITE LOCK ERROR
004752  076020                          EMT316
004754  076470                          EHT1
004756  076614                          EDT1
004760  076704                          EFT1

                                ;ERROR  317     HEADER CRC ERROR DURING DATA TRANSFER
004762  076032                          EMT317
004764  076470                          EHT1
004766  076614                          EDT1
004770  076704                          EFT1

                                ;ERROR  320     FORMAT ERROR DURING DATA TRANSFER
004772  076042                          EMT320
004774  076470                          EHT1
004776  076614                          EDT1
005000  076704                          EFT1

                                ;ERROR  321     HEADER COMPARE ERROR DURING DATA TRANSFER
005002  076052                          EMT321
005004  076470                          EHT1
005006  076614                          EDT1
005010  076704                          EFT1

                                ;ERROR  322     HEADER ERRORS SHOULD NOT BE SET
005012  076062                          EMT322
005014  076470                          EHT1
005016  076614                          EDT1
005020  076704                          EFT1
```

```
646
647                                     ;ERROR  323     DATA CHECK ERROR DURING DATA TRANSFER
648
    005022  076070                              EMT323
    005024  076470                              EHT1
    005026  076614                              EDT1
    005030  076704                              EFT1

649
650                                     ;ERROR  324     CORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
651
    005032  076100                              EMT324
    005034  076470                              EHT1
    005036  076614                              EDT1
    005040  076704                              EFT1

652
653                                     ;ERROR  325     UNCORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
654
    005042  076112                              EMT325
    005044  076470                              EHT1
    005046  076614                              EDT1
    005050  076704                              EFT1

655
656                                     ;ERROR  326     DATA PARITY ERROR DURING READ COMMMAND
657
    005052  076124                              EMT326
    005054  076470                              EHT1
    005056  076614                              EDT1
    005060  076704                              EFT1

658
659                                     ;ERROR  327     OFFSET MODE NOT RESET BY WRITE COMMAND
660
    005062  076142                              EMT327
    005064  076470                              EHT1
    005066  076614                              EDT1
    005070  076704                              EFT1

661
662                                     ;ERROR  330     DATA PARITY ERROR DURING WRITE COMMAND
663
    005072  076154                              EMT330
    005074  076470                              EHT1
    005076  076614                              EDT1
    005100  076704                              EFT1

664
665                                     ;ERROR  331     WRITE CLOCK FALURE DURING WRITE COMMAND
666
    005102  076164                              EMT331
    005104  076470                              EHT1
    005106  076614                              EDT1
    005110  076704                              EFT1

667
```

```
668                                     ;ERROR  332     DATA LATE ERROR DURING DATA TRANSFER
669
    005112  076176                                EMT332
    005114  076470                                EHT1
    005116  076614                                EDT1
    005120  076704                                EFT1

670
671                                     ;ERROR  333     PIP STIL SET AFTER DATA TRANSFER - SKI = 0
672
    005122  076210                                EMT333
    005124  076470                                EHT1
    005126  076614                                EDT1
    005130  076704                                EFT1

673
674                                     ;ERROR  334     LOST MOL DURING DATA TRANSFER - OPI = 0
675
    005132  076226                                EMT334
    005134  076470                                EHT1
    005136  076614                                EDT1
    005140  076704                                EFT1

676
677                                     ;ERROR  335     LOST VOLUME VALID DURING DATA TRANSFER - IVC  - 0
678
    005142  076244                                EMT335
    005144  076470                                EHT1
    005146  076614                                EDT1
    005150  076704                                EFT1

679
680                                     ;ERROR  336     DATA READ DOES NOT COMPARE WITH DATA WRITTEN
681
    005152  076264                                EMT336
    005154  076552                                EHT336
    005156  076650                                EDT336
    005160  076740                                EFT336

682
683                                     ;ERROR  337     WRITE CHECK ERROR NOT DETECTED
684
    005162  076274                                EMT337
    005164  076564                                EHT337
    005166  076660                                EDT337
    005170  076750                                EFT337

685
686                                     ;ERROR  340     WRITE CHECK ERROR AT UNEXPECTED ADDRESS
687
    005172  076304                                EMT340
    005174  076552                                EHT336
    005176  076650                                EDT336
    005200  076740                                EFT336

688
689                                     ;ERROR  341     INCORRECT DATA DURING WRITE CHECK ERROR
```

```
690
     005202  076316                          EMT341
     005204  076552                          EHT336
     005206  076650                          EDT336
     005210  076740                          EFT336

691
692                              ;ERROR  342     "IVC" ERROR NOT DETECTED DURING DATA TRANSFER
693
     005212  076324                          EMT342
     005214  076470                          EHT1
     005216  076614                          EDT1
     005220  076704                          EFT1

694
695                              ;ERROR  343     "FER" NOT DETECTED DURING DATA TRANSFER
696
     005222  076336                          EMT343
     005224  076470                          EHT1
     005226  076614                          EDT1
     005230  076704                          EFT1

697
698                              ;ERROR  344     "HCE" NOT DETECTED DURING DATA TRANSFER
699
     005232  076350                          EMT344
     005234  076576                          EHT344
     005236  076670                          EDT344
     005240  076760                          EFT344

700
701                              ;ERROR  345     "BSE" NOT DETECTED DURING DATA TRANSFER
702
     005242  076362                          EMT345
     005244  076470                          EHT1
     005246  076614                          EDT1
     005250  076704                          EFT1

703
704                              ;ERROR  346     HEADER ERROR WAS DETECTED W/ HCI SET
705
     005252  076372                          EMT346
     005254  076470                          EHT1
     005256  076614                          EDT1
     005260  076704                          EFT1

706
707                              ;ERROR  347     DATA TRANSFER NOT ABORTED W/ COMP ERROR SET
708
     005262  076406                          EMT347
     005264  076470                          EHT1
     005266  076614                          EDT1
     005270  076704                          EFT1

709
710                              ;ERROR  350     LOST VOLUME VALID DURING SEARCH - "IVC" = 0
711
```

```
005272  076420                          EMT350
005274  076470                          EHT1
005276  076614                          EDT1
005300  076704                          EFT1

712
713                             ;ERROR  351     "ATA" DID NOT SET DURING SEARCH
714
005302  076436                          EMT351
005304  076470                          EHT1
005306  076614                          EDT1
005310  076704                          EFT1

715
716                             ;ERROR  352     PROGRAM TIMEOUT WHILE TESTING RMLA
717
005312  076446                          EMT352
005314  000000                          0
005316  000000                          0
005320  000000                          0

718
719                             ;ERROR  353     LOOK AHEAD TEST FAILS
720
005322  076452                          EMT353
005324  076610                          EHT353
005326  076702                          EDT353
005330  076770                          EFT353

721
722                             ;ERROR  354     BSE SHOULD NOT BE SET
723
005332  076462                          EMT354
005334  076470                          EHT1
005336  076614                          EDT1
005340  076704                          EFT1

724
725                             ;PUT ERROR TABLE HERE
```

```
                        .SBTTL  ERROR TABLE USAGE

                        ;THE ERROR TABLE ABOVE CONSISTS OF FOUR WORD ENTRIES FOR EACH ERROR
                        ;NUMBER, I.E.,
                        ;
                        ;       EMT - ERROR MESSAGE TABLE ADDRESS
                        ;       EHT - ERROR HEADER TABLE ADDRESS
                        ;       EDT - ERROR DATA TABLE ADDRESS
                        ;       EFT - ERROR FORMAT TABLE ADDRESS
                        ;
                        ;THE EMT ENTRY IS THE ADDRESS OF THE TABLE OF ERROR MESSAGE STRINGS
                        ;FOR THE PARTICULAR ERROR.  EACH ERROR MESSAGE TABLE LISTS THE ADDRESS
                        ;OF ONE OR MORE ERROR MESSAGE STRINGS WHICH ARE TO BE FORMATTED AND
                        ;TYPED BY THE ERROR TYPE SUBROUTINE.  IF THE EMT ENTRY IS ZERO, THERE IS
                        ;NO MESSAGE TO BE TYPED FOR THE ERROR.
                        ;
                        ;SIMILARLY, THE EHT, EDT, AND EFT ENTRIES ARE ADDRESSES OF TABLES
                        ;OF HEADER, DATA AND FORMAT INFORMATION FOR A GIVEN ERROR.  EACH ENTRY
                        ;IN THE ERROR HEADER TABLE MAY OR MAY NOT HAVE AN ASSOCIATED LINE OF
                        ;DATA, HOWEVER, EACH DATA LINE MUST HAVE AN ASSOCIATED FORMAT AND
                        ;HEADER. THAT IS, A HEADER LINE MAY BE PRINTED WITHOUT ANY DATA,
                        ;BUT A DATA LINE IS NOT PRINTED WITHOUT A HEADER, AND EACH DATA LINE
                        ;MUST ALSO HAVE A FORMAT.
                        ;
                        ;IN SUMMARY,
                        ;       EACH NONZERO ENTRY IS THE ADDRESS OF A TABLE,
                        ;       EACH TABLE IS A LIST OF ADDRESSES WHICH DEFINES THE LOCATIONS
                        ;       OF MESSAGE STRINGS, HEADERS, DATA OR FORMAT.
```

```
                                        ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS
                                        
005342  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE THE TIME OUT OCCURED
005344  005740                          TST     -(R0)           ;ADJUST PC -2
005346  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
005350  104401  005356                  TYPE    ,65$            ;;TYPE ASCIZ STRING
005354  000417                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
005414                          64$:
005414  010046                          MOV     R0,-(SP)        ;SETUP FOR TYPING OUT PC
005416  104402                          TYPOC
005420  000240                          NOP                     ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
                                                                ;TO STOP ON UNEXPECTED TIMECUT.
                                        
                                        .SBTTL  START OF PROGRAM
                                        
005422  012737  177777  001330  START1: MOV     #-1,CHGADR      ;CHANGE RH/RM BUS ADDRESS
005430  000402                          BR      START2
                                        
005432  005037  001330          START:  CLR     CHGADR          ;NO CHANGE IN ADDRESS
005436  000240                  START2: NOP
005440  005227  000000                  INC     #0              ;TTY LOOP, WAIT FOR INCREMENT
005444  001375                          BNE     .-4             ;OF WORD
005446  000005                          RESET                   ;RESET THE WORLD
                                        
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
005450  012706  001114                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
005454  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
005456  022706  001154                  CMP     #SWR,R6         ;;DONE?
005462  001374                          BNE     .-6             ;;LOOP BACK IF NO
005464  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
005470  012737  064054  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005476  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005504  012737  064640  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINF
005512  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005520  012737  067674  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005526  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005534  012737  070002  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005542  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005550  013737  037100  037072          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
005556  005037  001206                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005562  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005566  112737  000001  001131          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005574  012737  005574  001122          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005602  012737  005602  001124          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
005610  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005614  012737  005650  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
005622  012737  177570  001154          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005630  012737  177570  001156          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005636  022777  177777  173310          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005644  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005646  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
```

```
        005650  012716  005656          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
        005654  000002                          RTI
        005656  012737  000176  001154  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
        005664  012737  000174  001156          MOV     #DISPREG,DISPLAY
        005672  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

        005676  005037  001230                  CLR     $PASS           ;;CLEAR PASS COUNT
        005702  132737  000200  001243          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
        005710  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
        005712  012737  001244  001154          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
        005720                          67$:
    24                                  ;SETUP ''TIMEOUT'' TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
    25  005720  012737  005342  000004          MOV     #BADTMO,ERRVEC  ;SETUP FOR UNEXPECTED TIMEOUT
    26  005726  012737  000300  000006          MOV     #PR6,ERRVEC+2   ;LEVEL 6
    27
    28                                  .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
        005734  005227  177777                  INC     #-1             ;;FIRST TIME?
        005740  001035                          BNE     68$             ;;BRANCH IF NO
        005742  022737  037242  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
        005750  001431                          BEQ     68$             ;;BRANCH IF YES
        005752  104401  005760                  TYPE    ,69$            ;;TYPE ASCIZ STRING
        005756  000426                          BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <CRLF>@CZRMOB0 - RM05/3/2 FUNCTIONAL TEST, PT 3@<CRLF>
        006034                          68$:
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
        006034  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
        006040  001012                          BNE     70$             ;;BRANCH IF YES
        006042  123727  001242  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
        006050  001406                          BEQ     70$             ;;BRANCH IF YES
        006052  023727  001154  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
        006060  001005                          BNE     71$             ;;BRANCH IF NO
        006062  104407                          GTSWR                   ;;GET SOFT-SWR SETTINGS
        006064  000403                          BR      71$
        006066  112737  000001  001150  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
        006074                          71$:
    29
    30                                  ;THE FOLLOWING FINDS OUT THE PROGRAM CONTROL MODE:
    31                                  ;PAPER TAPE (MANUAL), ACT11, XXDP CHAIN OR DUMP
    32
    33  006074  005037  001332                  CLR     XXDP            ;CLEAR 'XXDP' LOAD DEVICE STORAGE
    34  006100  122737  000016  000041          CMPB    #16,@#41        ;LOADED FROM AN RM05/3/2 ?
    35  006106  001160                          BNE     3$              ;BRANCH IF NOT
    36  006110  013737  000040  001332          MOV     @#40,XXDP       ;GET DEVICE INDICATOR AND NUMBER
    37  006116  122737  000007  001332          CMPB    #7,XXDP         ;IS IT A VALID NUMBER ?
    38  006124  103002                          BHIS    1$              ;YES
    39  006126  105037  001332                  CLRB    XXDP            ;NO, DEFAULT TO DRIVE 0
    40  006132  005737  000042          1$:     TST     @#42            ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
    41  006136  001425                          BEQ     2$              ;BR IF NEITHER
    42  006140  104401  006146                  TYPE    ,73$            ;;TYPE ASCIZ STRING
        006144  000412                          BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  <CRLF>/NOT TESTING DRIVE /
        006172                          72$:
    43  006172  005046                          CLR     -(SP)           ;CLEAR WORD ON STACK
    44  006174  113716  001332                  MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
    45  006200  104403                          TYPOS                   ;TYPE THE ADDRESS
    46  006202     001                          .BYTE   1               ;ONLY 1 CHARACTER
```

```
 47 006203    000                          .BYTE   0               ;SUPRESS LEADING ZEROS
 48 006204  104401  001217                 TYPE    ,$CRLF          ;CR-LF
 49 006210  000517                         BR      3$              ;GET NUMBER OF DRIVES
 50
 51 006212  005227  177777          2$:    INC     #-1             ;FIRST TIME THRU HERE ?
 52 006216  001114                         BNE     3$              ;NO
 53 006220  104401  006226                 TYPE    ,75$            ;;TYPE ASCIZ STRING
    006224  000410                         BR      74$             ;;GET OVER THE ASCIZ
                                    ;;75$:  .ASCIZ  <CRLF>/TO TEST DRIVE /
    006246                          74$:
 54 006246  005046                         CLR     -(SP)           ;CLEAR WORD ON STACK
 55 006250  113716  001332                 MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
 56 006254  104403                         TYPOS                   ;TYPE DRIVE ADDRESS
 57 006256     001                         .BYTE   1               ;ONLY 1 CHARACTER
 58 006257     000                         .BYTE   0               ;SUPRESS LEADING ZEROS
 59 006260  104401  006266                 TYPE    ,77$            ;;TYPE ASCIZ STRING
    006264  000431                         BR      76$             ;;GET OVER THE ASCIZ
                                    ;;77$:  .ASCIZ  /, HALT PROGRAM, REMOVE RRDP PACK AND REPLACE IT/<CRLF>
    006350                          76$:
 60 006350  104401  006356                 TYPE    ,78$            ;;TYPE ASCIZ STRING
    006354  000435                         BR      3$              ;;GET OVER THE ASCIZ
                                    ;;78$:  .ASCIZ  /WITH A WORK PACK, CLEAR LOCATION 40 AND RESTART PROGRAM./<CRLF>
    006450                          3$:
 64                                 ;CHECK FOR AUTO MODE OR STANDLONE MODE
 65 006450  005737  000042                 TST     @#42            ;RUNNING IN AUTO MODE ?
 66 006454  001561                         BEQ     STANDALONE      ;BR IF NO
 67 006456  012737  000377  001300         MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
 68
 69                                 ;PROGRAM IS RUNNING IN AUTO MODE - SEE IF SIZING IS ALLOWED
 70 006464                          XSIZ:
 71 006464  132737  000200  001243         BITB    #BIT7,$ENVM     ;SIZING ALLOWED ?
 72 006472  001146                         BNE     12$             ;NO
 73
 74 006474  005001                         CLR     R1              ;START FROM DRIVE 0
 75 006476  013700  001276                 MOV     $BASE,R0        ;LOAD THE BASE ADDRESS
 76 006502  104401  071430                 TYPE    ,SYSTAT         ;TYPE 'UNIT STATUS:'
 77
 78 006506  136137  071746  001300  1$:    BITB    ATNTBL(R1),$DEVM        ;IS DEVICE PRESENT IN MAP ?
 79 006514  001531                         BEQ     11$             ;BR IF NO
 80 006516  104401  001217                 TYPE    ,$CRLF          ;CR-LF
 81 006522  010146                         MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
    006524  104403                         TYPOS                   ;;GO TYPE--OCTAL ASCII
    006526     002                         .BYTE   2               ;;TYPE 2 DIGIT(S)
    006527     000                         .BYTE   0               ;;SUPPRESS LEADING ZEROS
 82 006530  104401  071640                 TYPE    ,BLNKS4         ;TYPE 4 BLANKS
 83
 84 006534  012760  000040  000010         MOV     #CLR,RMCS2(R0)  ;CLEAR MASS BUS
 85 006542  010160  000010                 MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
 86 006546  005760  000012                 TST     RMDS(R0)        ;ACCESS DRIVE REGISTER
 87 006552  032760  010000  000010         BIT     #NED,RMCS2(R0)  ;IS DRIVE PRESENT ?
 88 006560  001051                         BNE     3$              ;BR IF NO
 89 006562  032760  004000  000000         BIT     #DVA,RMCS1(R0)  ;IS DRIVE AVAILABLE ?
 90 006570  001450                         BEQ     4$              ;BR IF NO
 91 006572  016002  000026                 MOV     RMDT(R0),R2     ;SAVE DRIVE TYPE REGISTER IN R2
 92 006576  012737  071446  006776         MOV     #$RM02,10$      ;ASSUME RM02 DEVICE
 93 006604  022702  020025                 CMP     #20025,R2       ;SINGLE PORT RM02 ?
 94 006610  001430                         BEQ     2$              ;BR IF YES
```

```
 95 006612  022702  024025             CMP     #24025,R2       ;DUAL PORT RM02 ?
 96 006616  001425                     BEQ     2$              ;BR IF YES
 97 006620  012737  071453  006776     MOV     #$RM03,10$      ;ASSUME RM03 DEVICE
 98 006626  022702  020024             CMP     #20024,R2       ;SINGLE PORT RM03 ?
 99 006632  001417                     BEQ     2$              ;BR IF YES
100 006634  022702  024024             CMP     #24024,R2       ;DUAL PORT RM03 ?
101 006640  001414                     BEQ     2$              ;BR IF YES
102 006642  012737  071460  006776     MOV     #$RM05,10$      ;ASSUME RM05 DEVICE
103 006650  022702  020027             CMP     #20027,R2       ;SINGLE PORT RM05 ?
104 006654  001406                     BEQ     2$              ;BR IF YES
105 006656  022702  024027             CMP     #24027,R2       ;DUAL PORT RM05 ?
106 006662  001403                     BEQ     2$              ;BR IF YES
107 006664  104401  071465             TYPE    ,NOTRM          ;DRIVE NOT AN RM05/3/2
108 006670  000412                     BR      5$              ;CHECK NEXT DRIVE
109 006672  032760  010000  000012  2$:BIT     #MOL,RMDS(R0)   ;IS MEDIUM ON LINE ?
110 006700  001412                     BEQ     6$              ;BR IF NO
111 006702  000417                     BR      7$
112
113 006704  104401  071523          3$:TYPE    ,NOTPRS         ;DRIVE NOT PRESENT
114 006710  000402                     BR      5$              ;CHECK NFXT DRIVE
115
116 006712  104401  071540          4$:TYPE    ,NOTAVL         ;DRIVE NOT AVAILABLE
117 006716  146137  071746  001300  5$:BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
118 006724  000425                     BR      11$             ;CHECK NEXT DRIVE
119
120 006726  104401  071557          6$:TYPE    ,UNTOFF         ;DRIVE OFFLINE
121 006732  146137  071746  001300     BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
122 006740  000413                     BR      9$              ;PRINT DRIVE TYPE
123
124 006742  005737  001332          7$:TST     XXDP            ;LOADED FROM RM05/3/2 ?
125 006746  001406                     BEQ     8$              ;NO
126 006750  123701  001332             CMPB    XXDP,R1         ;IS THIS THE DRIVE ?
127 006754  001360                     BNE     5$              ;BR IF NO
128 006756  104401  071506             TYPE    ,LODEV          ;DRIVE IS LOAD DEVICE
129 006762  000755                     BR      5$
130
131 006764  104401  071570          8$:TYPE    ,UNTON          ;DRIVE ONLINE
132 006770  104401  071642          9$:TYPE    ,BLNKS2         ;TYPE 2 BLANKS
133 006774  104401                     TYPE                    ;PRINT DRIVE TYPE
134 006776  000000                 10$:.WORD   0               ;MESSAGE ADDRESS HERE
135
136 007000  005201                 11$:INC     R1              ;INCREMENT THE DRIVE ADDRESS
137 007002  020127  000007             CMP     R1,#7           ;ALL DRIVES ARE CHECKED ?
138 007006  003637                     BLE     1$              ;BRANCH IF NOT
139
140 007010  104401  001217         12$:TYPE    ,$CRLF          ;CR-LF
141 007014  000137  007476             JMP     CMNSTART        ;JUMP TO COMMON START
```

```
                                        .SBTTL  STANDALONE INPUT ROUTINES

007020                                  STANDALONE:
007020  004737  066310                          JSR     PC,$TKINT       ;INITIALIZE CONSOLE

007024  005227  177777                          INC     #-1             ;FIRST TIME THRU HERE ?
007030  001023                                  BNE     2$              ;BR IF NO

                                        ;SEE IF OPERATOR WANTS HELP TEXT

007032  104401  070530                          TYPE    ,MSHELP         ;WANT HELP ?
007036  104411                                  RDCHR                   ;GET RESPONSE
007040  012637  001176                          MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
007044  123727  001176  000131                  CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ?
007052  001005                                  BNE     1$              ;NO
007054  104401  001176                          TYPE    ,$TMP1          ;TYPE 'Y'
007060  104401  106436                          TYPE    ,HELP           ;YES - TYPE HELP TEXT
007064  000414                                  BR      3$
007066  104401  071634                  1$:     TYPE    ,N              ;TYPE 'N'
007072  104401  001217                          TYPE    ,$CRLF          ;CR-LF
007076  000407                                  BR      3$

                                        ;SEE IF USER WANTS TO CHANGE UNIBUS ADDRESS
007100                                  2$:
007100  005737  001330                          TST     CHGADR          ;CHANGE RH/RM BUS ADDRESS ?
007104  001457                                  BEQ     7$              ;BR IF NO
007106  005037  001330                          CLR     CHGADR          ;NO CHANGE NEXT TIME
007112  104401  001217                          TYPE    ,$CRLF          ;CR-LF

                                        ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, VECTOR ADDRESS AND INTERRUPT PRIORITY
007116                                  3$:
007116  104401  071105                          TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
007122  013746  001276                          MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
007126  104402                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
007130  104401  071642                          TYPE    ,BLNKS2         ;TYPE 2 BLANKS
007134  104413                                  RDOCT                   ;GET NEW BUS ADDRESS
007136  012637  001176                          MOV     (SP)+,$TMP1     ;CARRIAGE RETURN ?
007142  001412                                  BEQ     5$              ;YES-SKIP TO NEXT ENTRY
007144  022737  160000  001176                  CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE ?
007152  101403                                  BLOS    4$              ;YES
007154  104401  071115                          TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
007160  000756                                  BR      3$              ;TRY AGAIN
007162  013737  001176  001276          4$:     MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS

007170  104401  071157                  5$:     TYPE    ,CNSL03
007174  005046                                  CLR     -(SP)
007176  113716  001272                          MOVB    $VECT1,(SP)     ;GET CURRENT VECTOR ADDRESS
007202  104402                                  TYPOC
007204  104401  071642                          TYPE    ,BLNKS2         ;TYPE 2 BLANKS
007210  104413                                  RDOCT                   ;GET NEW VECTOR ADDRESS
007212  012637  001176                          MOV     (SP)+,$TMP1     ;CARRIAGE RETURN?
007216  001412                                  BEQ     7$              ;YES-SKIP TO NEXT ENTRY
007220  022737  0010C0  001176                  CMP     #1000,$TMP1     ;VECTOR ADDRESS < 1000 ?
007226  101003                                  BHI     6$              ;YES!!
007230  104401  071166                          TYPE    ,CNSL04         ;TYPE WARNING MESSAGE
007234  000755                                  BR      5$              ;RETRY
007236  113737  001176  001272          6$:     MOVB    $TMP1,$VECT1    ;STORE NEW VECTOR ADDRESS
```

```
                                        ;DIALOGUE TO INPUT DEVICE NUMBERS
007244  005227  177777          7$:     INC     #-1             ;FIRST TIME THRU ?
007250  001002                          BNE     8$              ;BR IF NO
007252  104401  071222                  TYPE    ,CNSL07         ;TYPE INPUT INSTRUCTIONS
007256  104401  001217          8$:     TYPE    ,$CRLF          ;CR-LF
007262  005037  001300          9$:     CLR     $DEVM           ;CLEAR DEVICE MAP
007266  104401  071406                  TYPE    ,MSDRVS         ;TYPE 'DRIVE(S): '
007272  104411                          RDCHR
007274  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
007300  023727  001176  000101          CMP     $TMP1,#'A       ;IS INPUT ''A'' ?
007306  001007                          BNE     10$             ;NO
007310  104401  070514                  TYPE    ,ALL            ;YES, TYPE ''ALL'' AND GO
007314  012737  000377  001300          MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
007322  000137  006464                  JMP     XSIZ            ;AUTO SIZE.

007326  023727  001176  000015  10$:    CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
007334  001436                          BEQ     12$             ;YES
007336  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
007342  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
007350  002430                          BLT     12$             ;YES
007352  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
007360  003427                          BLE     13$             ;NO
007362  000423                          BR      12$             ;ILLEGAL INPUT

007364  104411          11$:    RDCHR
007366  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
007372  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
007400  001432                          BEQ     14$             ;YES
007402  104401  070525                  TYPE    ,COMMA          ;TYPE ', '
007406  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
007412  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
007420  002404                          BLT     12$             ;YES
007422  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
007430  003403                          BLE     13$             ;NO
007432  104401  071364          12$:    TYPE    ,CNSL08         ;TYPE CR-LF '' ?ILLEGAL INPUT''
007436  000711                          BR      9$              ;RETRY

007440  013701  001176          13$:    MOV     $TMP1,R1        ;R1 = DRIVE NUMBER
007444  042701  177770                  BIC     #^C7,R1
007450  156137  071746  001300          BISB    ATNTBL(R1),$DEVM        ;SET DEVICE IN MAP
007456  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ?
007464  101337                          BHI     11$             ;NO
007466  104401  001217          14$:    TYPE    ,$CRLF          ;CR-LF
007472  000137  006464                  JMP     XSIZ            ;GO SIZE DEVICES
```

```
                                        ;ASSEMBLE TEST QUE FROM DEVICE MAP
007476                                  CMNSTART:
007476  104401  071600                          TYPE    ,DRIVES         ;TYPE 'DRIVE(S) TO BE TESTED'
007502  013700  001300                          MOV     $DEVM,R0        ;R0 = DEVICE MAP
007506  001004                                  BNE     1$              ;BR IF DRIVES TO TEST ?
007510  104401  070525                          TYPE    ,COMMA          ;TYPE ', '
007514  104401  071627                          TYPE    ,NONE           ;TYPE 'NONE'
007520  012701  001470                  1$:     MOV     #TSTQUE+2,R1    ;R1 = ADDRESS OF FIRST ENTRY IN QUE
007524  010137  001466                          MOV     R1,TSTQUE       ;INITIALIZE ENTRY POINTER
007530  012702  000001                          MOV     #1,R2           ;R2 = DEVICE POINTER
007534  005003                                  CLR     R3              ;R3 = DEVICE NUMBER
007536  030200                          2$:     BIT     R2,R0           ;IS THIS DEVICE IN MAP ?
007540  001413                                  BEQ     3$              ;NO .!
007542  104401  070525                          TYPE    ,COMMA          ;TYPE ', '
007546  010311                                  MOV     R3,(R1)         ;YES - ENTER DEVICE NUMBER IN QUE
007550  010346                                  MOV     R3,-(SP)        ;;SAVE R3 FOR TYPEOUT
007552  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
007554     001                                  .BYTE   1               ;;TYPE 1 DIGIT(S)
007555     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
007556  116361  071746  000001                  MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
007564  062701  000002                          ADD     #2,R1           ;ADVANCE ENTRY POINTER
007570  006302                          3$:     ASL     R2              ;ADVANCE DEVICE POINTER
007572  105702                                  TSTB    R2              ;DONE ALL DEVICES ?
007574  001402                                  BEQ     4$              ;YES
007576  005203                                  INC     R3              ;ADVANCE DEVICE NUMBER
007600  000756                                  BR      2$              ;ENTER NEXT DEVICE
007602  005011                          4$:     CLR     (R1)            ;TERMINATE TEST QUE
007604  104401  001217                          TYPE    ,$CRLF          ;TYPE CR-LF

                                        ;SIZE FOR CLOCK
007610  004737  043550                          JSR     PC,SIZCLK       ;SEE IF CLOCK PRESENT
007614  000425                                  BR      6$              ;YES - CLOCK IS PRESENT
007616  104401  007624                          TYPE    ,65$            ;;TYPE ASCIZ STRING
007622  000413                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF>/NO 'L' OR 'P' CLOCK/
007652                                  64$:
007652  005737  000042                          TST     @#42            ;ANY MONITOR PRESENT ?
007656  001002                                  BNE     5$              ;BR IF YES
007660  000137  005432                          JMP     START           ;JUMP TO START
007664  000137  037232                  5$:     JMP     $GET42          ;RETURN CONTROL TO MONITOR
007670                                  6$:

007670  000240                          READY:  NOP                     ;READY TO START TEST
007672  105737  001300                          TSTB    $DEVM           ;ANY DRIVES IN MAP ?
007676  001007                                  BNE     2$              ;BR IF YES
007700  005737  000042                          TST     @#42            ;ANY MONITOR PRESENT ?
007704  001002                                  BNE     1$              ;BR IF YES
007706  000137  005432                          JMP     START           ;JUMP TO START
007712  000137  037232                  1$:     JMP     $GET42          ;RETURN CONTROL TO MONITOR

007716  105037  001116                  2$:     CLRB    $TSTNM          ;RESET TEST NUMBER
007722  005037  001206                          CLR     $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
007726  005037  001326                          CLR     CTLFG           ;CLEAR CONTROL-C FLAG
007732  004737  066310                          JSR     PC,$TKINT       ;INITIALIZE TTY
007736  012746  000240                          MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
007742  012746  007750                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
007746  000002                                  RTI                     ;;POP NEW PC AND PS
```

```
                007750                          64$:
     50 007750  117737  171512  001234                 MOVB    @TSTQUE,$UNIT   ;LOAD DRIVE NUMBER
     51 007756  005037  001512                         CLR     MEDENB          ;CLEAR MEDIA ENABLE
     52
     53                                         ;CLEAR MASSBUS CONTROLLER, SELECT DRIVE AND DETERMINE THE LAST TRACK
     54                                         ;OF THE DIFFERENT DRIVE TYPES
     55
     56 007762  012737  002000  001334                 MOV     #TA4,LSTRK      ;ASSUME LAST TRACK FOR RM02/3 = 4.
     57 007770  013700  001276                         MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     58 007774  012760  000040  000010                 MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
     59 010002  117760  171460  000010                 MOVB    @TSTQUE,RMCS2(R0)       ;SELECT DEVICE UNDER TEST
     60 010010  016002  000026                         MOV     RMDT(R0),R2     ;GET RMDT AND
     61 010014  042702  177770                         BIC     #177770,R2      ;SAVE DRIVE TYPE BITS
     62 010020  022702  000007                         CMP     #7,R2           ;IS IT AN RM05 ?
     63 010024  001003                                 BNE     3$              ;NO, MUST BE AN RM02 OR RM03
     64 010026  012737  011000  001334                 MOV     #TA16.TA2,LSTRK ;YES--SET LAST TRACK - 18.
     65
     66                                         ;TYPE DRIVE NUMBER TO BE TESTED($UNIT)
     67
     68 010034  104401  001217                 3$:     TYPE    ,$CRLF          ;CR-LF
     69 010040  104401  071422                         TYPE    ,MSGDRV         ;TYPE 'DRIVE'
     70 010044  013746  001234                         MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                                ;;TYPE DRIVE NUMBER
        010050  104403                                 TYPOS                   ;;GO TYPE--OCTAL ASCII
        010052     002                                 .BYTE   2               ;;TYPE 2 DIGIT(S)
        010053     000                                 .BYTE   0               ;;SUPPRESS LEADING ZEROS
     71 010054  005004                                 CLR     R4              ;THESE TWO LOOPS ARE ADDED TO
     72 010056  005304                                 DEC     R4              ;WAIT FOR TTY
     73 010060  001376                                 BNE     .-2
     74 010062  005304                                 DEC     R4
     75 010064  001376                                 BNE     .-2
```

```
                                ;;*****************************************************************
                                ;*TEST 1        CONTROLLER ACCESS TEST
                                ;;*****************************************************************
                                ;
010066                          TST1:
010066  000004                          SCOPE                   ;SCOPE CALL
010070  000240                          NOP                     ;START OF TEST
010072  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
010076  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
010102  013701  001466                  MOV     TS~QUE,R1       ;(R1) = DEVICE BEING TESTED
010106  012737  000001  001226          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                        
010114  005001                          CLR     R1
010116  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
010122  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
010126  012737  010230  000004          MOV     #1$,ERRVEC
010134  012737  000300  000006          MOV     #PR6,ERRVEC+2
                        
010142  110160  000001                  MOVB    R1,RMCS1+1(R0)  ;MOVE HI BYTE TO RMCS1
010146  010160  000002                  MOV     R1,RMWC(R0)     ;MOVE WORD COUNT REGISTER
010152  016002  000002                  MOV     RMWC(R0),R2
010156  010160  000004                  MOV     R1,RMBA(R0)     ;MOVE BUS ADDRESS REGISTER
010162  016002  000004                  MOV     RMBA(R0),R2
010166  016046  000010                  MOV     RMCS2(R0),-(SP)         ;;PUSH RMCS2(R0) ON STACK
010172  010160  000010                  MOV     R1,RMCS2(R0)    ;MOVE CONTROL STATUS REGISTER
010176  016002  000010                  MOV     RMCS2(R0),R2
010202  012660  000010                  MOV     (SP)+,RMCS2(R0)         ;;POP STACK INTO RMCS2(R0)
010206  010160  000022                  MOV     R1,RMDB(R0)     ;MOVE DATA BUFFER
010212  016002  000022                  MOV     RMDB(R0),R2
010216  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
010222  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
010226  000417                          BR      3$              ;NO BUS TIMEOUT OCCURRED
                        
010230  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
010232  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
010236  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
010242  104110                          EMT     110
010244  005737  000042                  TST     @#42            ;STAND ALONE MODE ?
010250  001002                          BNE     2$              ;NO!!
010252  000137  005422                  JMP     START1          ;YES-GO GET $BASE
010256  005037  001300          2$:     CLR     $DEVM           ;FUDGE NO DRIVES IN MAP
010262  000137  037044                  JMP     $EOP            ;RETURN TO $EOP
010266                          3$:
                        
                                ;;*****************************************************************
                                ;*TEST 2        WRITE, READ ZEROS
                                ;;*****************************************************************
010266                          TST2:
010266  000004                          SCOPE                   ;SCOPE CALL
010270  000240                          NOP                     ;START OF TEST
010272  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
010276  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
010302  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
010306  012737  000002  001226          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                        
                                ;;*****************************************************************
                                ;LOOP #1        FORMAT,WRITE,READ
```

```
                                        1$:

                                        ;PREPARE DEVICE FOR TEST
010314  004737  037316                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
010320  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
010322  000404                                BR      2$              ;GO TO 2$ IF NO ERROR
010324  000240                                NOP                     ;RETURN HERE IF ERROR
010326  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
010330  000137  011274                        JMP     25$             ;GO TO 25$ IF ERROR

                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
010334                                  2$:
010334  012737  000000  001446                MOV     #0,RMDC0                ;CYLINDER = 0
010342  012737  000000  001420                MOV     #0,RMDA0                ;TRACK - 0, SECTOR   0
010350  012737  106436  001416                MOV     #BUFONE,RMBA0           ;BUS ADDRESS
010356  012737  177376  001414                MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
010364  012737  010000  001444                MOV     #FMT16,RMOF0            ;16 BIT FORMAT
010372  012737  000063  001412                MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND

                                        ;VERIFY THAT SECTOR IS NOT BAD
010400  004737  040242                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
010404  000405                                BR      3$              ;GO TO 3$ IF NO ERROR
010406  104401  070426                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
010412  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
010414  000137  011274                        JMP     25$             ;GO TO 25$ IF ERROR
010420                                  3$:
010420  012737  072060  001174                MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
010426  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
010434  004737  042174                        JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
010440  012702  001553                        MOV     #PUTINX,R2              ;R2   BYTE ENTRY POSITION
010444  112722  000034                        MOVB    #RMDC,(R2)+
010450  112722  000006                        MOVB    #RMDA,(R2)+
010454  112722  000004                        MOVB    #RMBA,(R2)+
010460  112722  000002                        MOVB    #RMWC,(R2)+
010464  112722  000032                        MOVB    #RMOF,(R2)+
010470  112722  000000                        MOVB    #RMCS1,(R2)+
010474  112712  000200                        MOVB    #200,(R2)               ;TERMINATE TABLE
010500                                  4$:
010500  004737  043330                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
010504  000404                                BR      5$              ;GO TO 5$ IF NO ERROR
010506  000240                                NOP                     ;RETURN HERE IF ERROR
010510  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
010512  000137  011274                        JMP     25$             ;GO TO 25$ IF ERROR
010516                                  5$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
010516  004737  042774                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
```

```
                                        ;WAIT FOR COMMAND TO COMPLETE
 010522  004737  043672                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
 010526  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
 010532  000404                                  BR      6$              ;GO TO 6$ IF NO ERROR
 010534  000240                                  NOP                     ;RETURN HERE IF ERROR
 010536  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
 010540  000137  011274                          JMP     25$             ;GO TO 25$ IF ERROR
 010544                                  6$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
 010544  004737  056572                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
 010550  000405                                  BR      7$              ;GO TO 7$ IF NO ERROR
 010552  000240                                  NOP                     ;RETURN HERE IF ERROR
 010554  104000                                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
 010556  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 010560  000137  011274                          JMP     25$             ;GO TO 25$ IF ERROR
 010564                                  7$:

                                        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 010564  012737  010574  001124                  MOV     #8$,$LPERR
 010572  000410                                  BR      9$                      ;SKIP TO WRITE OPERATION

                                        ;;*****************************************************************
                                        ;LOOP #2        WRITE,READ

 010574                                  8$:

                                        ;PREPARE DEVICE FOR TEST
 010574  004737  037316                  ;       JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
 010600  154130                          ;       .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                        ;                               ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                        ;                               ;CLEAR CONTROLLER & SELECT DEVICE
                                        ;                               ;VERIFY CONTROLLER CLEAR OPERATION
                                        ;                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                        ;                               ;VERIFY PACK ACKNOWLEDGE
                                        ;                               ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                        ;                               ;VERIFY RECALIBRATION
 010602  000404                          ;       BR      9$              ;GO TO 9$ IF NO ERROR
 010604  000240                          ;       NOP                     ;RETURN HERE IF ERROR
 010606  104000                          ;       EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
 010610  000137  011274                  ;       JMP     25$             ;GO TO 25$ IF ERROR
 010614                                  9$:

                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
 010614                                  10$:
 010614  012737  106442  001416                  MOV     #BUFONE+4,RMBAO         ;MOVE MEMORY ADDRESS
 010622  012737  177400  001414                  MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
 010630  012737  000061  001412                  MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
 010636  012702  001553                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 010642  112722  000006                          MOVB    #RMDA,(R2)+
 010646  112722  000034                          MOVB    #RMDC,(R2)+
 010652  112722  000032                          MOVB    #RMOF,(R2)+
 010656  112722  000004                          MOVB    #RMBA,(R2)+
 010662  112722  000002                          MOVB    #RMWC,(R2)+
```

```
                                                MOVB    #RMCS1,(R2)+
                                                MOVB    #200,(R2)+              ;TERMINATE TABLE

                                                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                                BR      1$              ;GO TO 11$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                                JMP     25$             ;GO TO 25$ IF ERROR
                                        11$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
                                                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
                                                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
                                                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                BR      12$             ;GO TO 12$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                                JMP     25$             ;GO TO 25$ IF ERROR
                                        12$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
                                                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                                                BR      13$             ;GO TO 13$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                                JMP     25$             ;GO TO 25$ IF ERROR
                                        13$:
                                                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                                BR      14$             ;GO TO 14$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                                JMP     25$             ;GO TO 25$ IF ERROR
                                        14$:
                                                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                                                BR      15$             ;GO TO 15$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                                JMP     25$             ;GO TO 25$ IF ERROR
                                        15$:

                                        ;CHANGE LOOP ADDRESSES
                                                MOV     #16$,$LPERR
                                                BR      17$                             ;SKIP TO NEXT OPERATION

                                        ;;*****************************************************************************
                                        ;LOOP #3        READ

                                        16$:

                                        ;PREPARE DEVICE FOR TEST
```

```
    139 011032  004737  037316          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        011036  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                ;VERIFY RECALIBRATION
        011040  000404                  BR      17$             ;GO TO 17$ IF NO ERROR
        011042  000240                  NOP                     ;RETURN HERE IF ERROR
        011044  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        011046  000137  011274          JMP     25$             ;GO TO 25$ IF ERROR
    140 011052                  17$:
    141
    142                             ;SETUP PARAMETERS AND EXCUTE COMMAND
    143 011052                  18$:
    144 011052  012737  107446  001416        MOV     #BUFTWO+4,RMBAO         ;CHANGE MEMORY ADDRESS
    145 011060  012737  000071  001412        MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
    146 011066  012702  001553              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    147 011072  112722  000006              MOVB    #RMDA,(R2)+
    148 011076  112722  000032              MOVB    #RMOF,(R2)+
    149 011102  112722  000034              MOVB    #RMDC,(R2)+
    150 011106  112722  000004              MOVB    #RMBA,(R2)+
    151 011112  112722  000002              MOVB    #RMWC,(R2)+
    152 011116  112722  000000              MOVB    #RMCS1,(R2)+
    153 011122  112712  000200              MOVB    #200,(R2)
    154
    155 011126  004737  043330          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        011132  000404                  BR      19$             ;GO TO 19$ IF NO ERROR
        011134  000240                  NOP                     ;RETURN HERE IF ERROR
        011136  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        011140  000137  011274          JMP     25$             ;GO TO 25$ IF ERROR
    156 011144                  19$:
    157
    158                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        011144  004737  042774          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    159
    160                             ;WAIT FOR COMMAND TO COMPLETE
        011150  004737  043672          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    161
    162                             ;GO GET REGISTER STATUS
    163 011154  004737  043060          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        011160  000404                  BR      20$             ;GO TO 20$ IF NO ERROR
        011162  000240                  NOP                     ;RETURN HERE IF ERROR
        011164  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        011166  000137  011274          JMP     25$             ;GO TO 25$ IF ERROR
    164 011172                  20$:
    165
    166                             ;VERIFY RESULTS OF READ COMMAND
    167 011172  004737  044056          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        011176  000405                  BR      21$             ;GO TO 21$ IF NO ERROR
        011200  000240                  NOP                     ;RETURN HERE IF ERROR
        011202  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        011204  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011206  000137  011274          JMP     25$             ;GO TO 25$ IF ERROR
    168 011212                  21$:
```

```
    169 011212  004737  056572              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        011216  000405                      BR      22$             ;GO TO 22$ IF NO ERROR
        011220  000240                      NOP                     ;RETURN HERE IF ERROR
        011222  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        011224  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011226  000137  011274              JMP     25$             ;GO TO 25$ IF ERROR
    170 011232                          22$:
    171 011232  004737  044710              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        011236  000405                      BR      23$             ;GO TO 23$ IF NO ERROR
        011240  000240                      NOP                     ;RETURN HERE IF ERROR
        011242  104000                      EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        011244  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011246  000137  011274              JMP     25$             ;GO TO 25$ IF ERROR
    172 011252                          23$:
    173
    174                                 ;GO VERIFY DATA
    175 011252  004737  042432              JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        011256  106442                      .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        011260  107446                      .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        011262  000404                      BR      24$             ;GO TO 24$ IF NO ERROR
        011264  000240                      NOP                     ;RETURN HERE IF ERROR
        011266  104000                      EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        011270  000137  011274              JMP     25$             ;GO TO 25$ IF ERROR
    176 011274                          24$:
    177
    178 011274                          25$:
    179
    180                                 ;;*******************************************************************
                                        ;*TEST 3         WRITE, WRITE CHECK ZEROS
                                        ;;*******************************************************************
        011274                          TST3:
        011274  000004                      SCOPE                   ;SCOPE CALL
        011276  000240                      NOP                     ;START OF TEST
        011300  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        011304  013700  001276              MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        011310  013701  001466              MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        011314  012737  000003  001226      MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
    181
    182                                 ;;*******************************************************************
    183                                 ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
    184
    185 011322                          1$:
    186
    187                                 ;PREPARE DEVICE FOR TEST
    188 011322  004737  037316              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        011326  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
        011330  000404                      BR      2$              ;GO TO 2$ IF NO ERROR
        011332  000240                      NOP                     ;RETURN HERE IF ERROR
        011334  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        011336  000137  012252              JMP     24$             ;GO TO 24$ IF ERROR
```

```
                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
011342                                  2$:
011342  012737  000000  001446                MOV     #0,RMDCO                ;CYLINDER = 0
011350  012737  000000  001420                MOV     #0,RMDAO                ;TRACK = 0, SECTOR - 0
011356  012737  106436  001416                MOV     #BUFONE,RMBAO           ;BUS ADDRESS
011364  012737  177376  001414                MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
011372  012737  010000  001444                MOV     #FMT16,RMOFO            ;16 BIT FORMAT
011400  012737  000063  001412                MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND

                                        ;VERIFY THAT SECTOR IS NOT BAD
011406  004737  040242                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
011412  000405                            BR      3$              ;GO TO 3$ IF NO ERROR
011414  104401  070426                    TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
011420  104000                            EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
011422  000137  012252                    JMP     24$             ;GO TO 24$ IF ERROR
011426                                  3$:
011426  012737  072060  001174                MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
011434  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
011442  004737  042174                      JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
011446  012702  001553                      MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
011452  112722  000034                      MOVB    #RMDC,(R2)+
011456  112722  000006                      MOVB    #RMDA,(R2)+
011462  112722  000004                      MOVB    #RMBA,(R2)+
011466  112722  000002                      MOVB    #RMWC,(R2)+
011472  112722  000032                      MOVB    #RMOF,(R2)+
011476  112722  000000                      MOVB    #RMCS1,(R2)+
011502  112712  000200                      MOVB    #200,(R2)               ;TERMINATE TABLE
011506                                  4$:
011506  004737  043330                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011512  000404                            BR      5$              ;GO TO 5$ IF NO ERROR
011514  000240                            NOP                     ;RETURN HERE IF ERROR
011516  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
011520  000137  012252                    JMP     24$             ;GO TO 24$ IF ERROR
011524                                  5$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
011524  004737  042774                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
011530  004737  043672                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
011534  004737  043060                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
011540  000404                            BR      6$              ;GO TO 6$ IF NO ERROR
011542  000240                            NOP                     ;RETURN HERE IF ERROR
011544  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
011546  000137  012252                    JMP     24$             ;GO TO 24$ IF ERROR
011552                                  6$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
011552  004737  056572                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
011556  000405                            BR      7$              ;GO TO 7$ IF NO ERROR
011560  000240                            NOP                     ;RETURN HERE IF ERROR
011562  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
```

```
        011564  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011566  000137  012252                  JMP     24$             ;GO TO 24$ IF ERROR
    228 011572                          7$:
    229
    230                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
    231 011572  012737  011602  001124          MOV     #8$,$LPERR
    232 011600  000410                          BR      9$                      ;SKIP TO WRITE OPERATION
    233
    234                                 ;;*****************************************************************
    235                                 ;LOOP #2        WRITE,WRITE CHECK DATA
    236
    237 011602                          8$:
    238
    239                                 ;PREPARE DEVICE FOR TEST
    240 011602  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        011606  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        011610  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
        011612  000240                          NOP                     ;RETURN HERE IF ERROR
        011614  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        011616  000137  012252                  JMP     24$             ;GO TO 24$ IF ERROR
    241 011622                          9$:
    242
    243                                 ;SETUP PARAMETERS AND EXCUTE COMMAND
    244 011622                          10$:
    245 011622  012737  106442  001416          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
    246 011630  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
    247 011636  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
    248 011644  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    249 011650  112722  000006                  MOVB    #RMDA,(R2)+
    250 011654  112722  000034                  MOVB    #RMDC,(R2)+
    251 011660  112722  000032                  MOVB    #RMOF,(R2)+
    252 011664  112722  000004                  MOVB    #RMBA,(R2)+
    253 011670  112722  000002                  MOVB    #RMWC,(R2)+
    254 011674  112722  000000                  MOVB    #RMCS1,(R2)+
    255 011700  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
    256
    257 011704  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        011710  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
        011712  000240                          NOP                     ;RETURN HERE IF ERROR
        011714  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        011716  000137  012252                  JMP     24$             ;GO TO 24$ IF ERROR
    258 011722                          11$:
    259
    260                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        011722  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    261
    262                                 ;WAIT FOR COMMAND TO COMPLETE
        011726  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    263
    264                                 ;GO GET REGISTER STATUS
```

```
                                ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                ;GO TO 12$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY GET SUBROUTINE
                                ;GO TO 24$ IF ERROR
12$:

;VERIFY RESULTS OF WRITE COMMAND
                                ;GO CHECK FOR PRIMARY FRROPS
                                ;GO TO 13$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                ;GO BACK FOR MORE ERROR CHECKS
                                ;GO TO 24$ IF ERROR
13$:
                                ;GO VERIFY RESULTS OF DATA TRANSFER
                                ;GO TO 14$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                ;GO BACK FOR MORE ERROR CHECKS
                                ;GO TO 24$ IF ERROR
14$:
                                ;GO CHECK FOR SECONDARY ERRORS
                                ;GO TO 15$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY SECERR SUBROUTINE
                                ;GO BACK FOR MORE ERROR CHECKS
                                ;GO TO 24$ IF ERROR
15$:

;CHANGE LOOP ADDRESSES
                                ;SKIP TO NEXT OPERATION

;;*****************************************************************
;LOOP #3         WRITE CHECK DATA

16$:

;PREPARE DEVICE FOR TEST
                                ;PREPARE DEVICE FOR TEST
                                ;TASK DESCRIPTOR AS FOLLOWS:
                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                ;CLEAR CONTROLLER & SELECT DEVICE
                                ;VERIFY CONTROLLER CLEAR OPERATION
                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                ;VERIFY PACK ACKNOWLEDGE
                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                ;VERIFY RECALIBRATION
                                ;GO TO 17$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                                ;GO TO 24$ IF ERROR
17$:

;SETUP PARAMETERS AND EXCUTE COMMAND
18$:
```

```
                                                ;WRITE CHECK DATA DATA COMMAND
                                                ;LOAD PUT REGISTER INDEX TABLE
                                                ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                                ;GO TO 19$ IF NO ERROR
                                                ;RETURN HERE IF ERROR
                                                ;ERROR # DEFINED BY PUT SUBROUTINE
                                                ;GO TO 24$ IF ERROR
                                                ;GO TO GETSTS SUBROUTINE
                                                ;GO TO TIMOUT SUBROUTINE
                                                ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                ;GO TO 20$ IF NO ERROR
                                                ;RETURN HERE IF ERROR
                                                ;ERROR # DEFINED BY GET SUBROUTINE
                                                ;GO TO 24$ IF ERROR
                                                ;GO CHECK FOR PRIMARY ERRORS
                                                ;GO TO 21$ IF NO ERROR
                                                ;RETURN HERE IF ERROR
                                                ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                                ;GO BACK FOR MORE ERROR CHECKS
                                                ;GO TO 24$ IF ERROR
                                                ;GO VERIFY RESULTS OF DATA TRANSFER
                                                ;GO TO 22$ IF NO ERROR
                                                ;RETURN HERE IF ERROR
                                                ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                                ;GO BACK FOR MORE ERROR CHECKS
                                                ;GO TO 24$ IF ERROR
                                                ;GO CHECK FOR SECONDARY ERRORS
                                                ;GO TO 23$ IF NO ERROR
                                                ;RETURN HERE IF ERROR
                                                ;ERROR # DEFINED BY SECERP SUBROUTINE
                                                ;GO BACK FOR MORE ERROR CHECKS
                                                ;GO TO 24$ IF ERROR
```

```
                                        ;*TEST 4        WRITE, WRITE CHECK ZEROS W/ WCE ERROR
                                        ;;*****************************************************************
012252                                  *ST4:
012252  000004                                SCOPE                   ;SCOPE CALL
012254  000240                                NOP                     ;START OF TEST
012256  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
012262  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
012266  013701  001466                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
012272  012737  000004  001226                MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
323
324                                     ;;*****************************************************************
325                                     ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
326
327 012300                              1$:
328
329                                     ;PREPARE DEVICE FOR TEST
330 012300  004737  037316                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    012304  154130                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
    012306  000404                              BR      2$              ;GO TO 2$ IF NO ERROR
    012310  000240                              NOP                     ;RETURN HERE IF ERROR
    012312  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    012314  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
331
332                                     ;LOAD PARAMETERS AND GENERATE DATA BUFFER
333 012320                              2$:
334 012320  012737  000000  001446              MOV     #0,RMDC0                ;CYLINDER = 0
335 012326  012737  000000  001420              MOV     #0,RMDA0                ;TRACK = 0, SECTOR   0
336 012334  012737  106436  001416              MOV     #BUFONE,RMBA0           ;BUS ADDRESS
337 012342  012737  177376  001414              MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
338 012350  012737  010000  001444              MOV     #FMT16,RMOF0            ;16 BIT FORMAT
339 012356  012737  000063  001412              MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
340
341                                     ;VERIFY THAT SECTOR IS NOT BAD
    012364  004737  040242                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    012370  000405                              BR      3$              ;GO TO 3$ IF NO ERROR
    012372  104401  070426                      TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    012376  104000                              EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    012400  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
342 012404                              3$:
343 012404  012737  072060  001174              MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
344 012412  012737  000001  001176              MOV     #1,$TMP1                ;RANGE OF PATTERN
345 012420  004737  042174                      JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
346
347 012424  012702  001553                      MOV     #PUTINX,R2              ;R2 - BYTE ENTRY POSITION
348 012430  112722  000034                      MOVB    #RMDC,(R2)+
349 012434  112722  000006                      MOVB    #RMDA,(R2)+
350 012440  112722  000004                      MOVB    #RMBA,(R2)+
351 012444  112722  000002                      MOVB    #RMWC,(R2)+
352 012450  112722  000032                      MOVB    #RMOF,(R2)+
353 012454  112722  000000                      MOVB    #RMCS1,(R2)+
```

```
                                        MOVB    #200,(R2)               ;TERMINATE TABLE
                                4$:
                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      5$              ;GO TO 5$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     29$             ;GO TO 29$ IF ERROR
                                5$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
                                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                ;WAIT FOR COMMAND TO COMPLETE
                                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                ;GO EGT REGISTER STATUS
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      6$              ;GO TO 6$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     29$             ;GO TO 29$ IF ERROR
                                6$:

                                ;VERIFY RESULTS OF WRITE COMMAND
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      7$              ;GO TO 7$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     29$             ;GO TO 29$ IF ERROR
                                7$:

                                ;MOVE LOOP ADDRESSES TO NEXT OPERATION
                                        MOV     #8$,$LPERR
                                        BR      9$                      ;SKIP TO WRITE OPERATION

                                ;;*****************************************************************
                                ;LOOP #2        WRITE,WRITE CHECK DATA

                                8$:

                                ;PREPARE DEVICE FOR TEST
                                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
                                        BR      9$              ;GO TO 9$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                                        JMP     29$             ;GO TO 29$ IF ERROR
                                9$:
```

```
                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
012600                                  10$:
012600  012737  177400  001414                MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
012606  012737  106442  001416                MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
012614  012737  000061  001412                MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
012622  012702  001553                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
012626  112722  000006                        MOVB    #RMDA,(R2)+
012632  112722  000034                        MOVB    #RMDC,(R2)+
012636  112722  000032                        MOVB    #RMOF,(R2)+
012642  112722  000004                        MOVB    #RMBA,(R2)+
012646  112722  000002                        MOVB    #RMWC,(R2)+
012652  112722  000000                        MOVB    #RMCS1,(R2)+
012656  112722  000200                        MOVB    #200,(R2)+              ;TERMINATE TABLE

012662  004737  043330                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
012666  000404                              BR      11$             ;GO TO 11$ IF NO ERROR
012670  000240                              NOP                     ;RETURN HERE IF ERROR
012672  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
012674  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
012700                                  11$:


                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
012700  004737  042774                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
012704  004737  043672                        JSR     PC,TIMOUT       ;GC TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE COMMAND
012710  004737  043060                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012714  000404                              BR      12$             ;GO TO 12$ IF NO ERROR
012716  000240                              NOP                     ;RETURN HERE IF ERROR
012720  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
012722  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
012726                                  12$:


                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
012726  004737  044056                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
012732  000405                              BR      13$             ;GO TO 13$ IF NO ERROR
012734  000240                              NOP                     ;RETURN HERE IF ERROR
012736  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
012740  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012742  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
012746                                  13$:
012746  004737  056572                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
012752  000405                              BR      14$             ;GO TO 14$ IF NO ERROR
012754  000240                              NOP                     ;RETURN HERE IF ERROR
012756  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
012760  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012762  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
012766                                  14$:
012766  004737  044710                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
012772  000405                              BR      15$             ;GO TO 15$ IF NO ERROR
012774  000240                              NOP                     ;RETURN HERE IF ERROR
012776  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
013000  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013002  000137  013446                      JMP     29$             ;GO TO 29$ IF ERROR
013006                                  15$:
```

```
                                        ;CHANGE LOOP ADDRESSES
013006  012737  013014  001124          MOV     #16$,$LPERR

                                        ;;*****************************************************************
                                        ;LOOP #3        WRITE CHECK DATA

013014                                  16$:

013014  012703  000001                  MOV     #1,R3                   ;R3+WCE BIT POSITION
013020  050337  107440          17$:    BIS     R3,BUFTWO-2             ;CHANGE LAST WORD OF BUFFER

                                        ;PREPARE DEVICE FOR TEST
013024  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
013030  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                ;VERIFY RECALIBRATION
013032  000404                          BR      18$             ;GO TO 18$ IF NO ERROR
013034  000240                          NOP                     ;RETURN HERE IF ERROR
013036  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
013040  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR

013044                                  18$:

                                        ;READ DATA FROM DEVICE
013044                                  19$:
013044  012737  000051  001412          MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
013052  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
013056  112722  000006                  MOVB    #RMDA,(R2)+
013062  112722  000032                  MOVB    #RMOF,(R2)+
013066  112722  000034                  MOVB    #RMDC,(R2)+
013072  112722  000004                  MOVB    #RMBA,(R2)+
013076  112722  000002                  MOVB    #RMWC,(R2)+
013102  112722  000000                  MOVB    #RMCS1,(R2)+
013106  112712  000200                  MOVB    #200,(R2)

013112  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
013116  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
013120  000240                          NOP                     ;RETURN HERE IF ERROR
013122  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
013124  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR

013130                                  20$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
013130  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
013134  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
013140  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
013144  000404                          BR      21$             ;GO TO 21$ IF NO ERROR
013146  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        013150  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        013152  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR
    453 013156                          21$:
    454
    455                                 ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
    456 013156  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        013162  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
        013164  000240                          NOP                     ;RETURN HERE IF ERROR
        013166  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        013170  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013172  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR
    457 013176                          22$:
    458 013176  032737  040000  001346          BIT     #WCE,RMCS2I             ;WAS 'WCE' DETECTED??
    459 013204  001030                          BNE     24$                     ;YES!!
    460
    461 013206  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        013212  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
        013214  000240                          NOP                     ;RETURN HERE IF ERROR
        013216  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        013220  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013222  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR
    462 013226                          23$:
    463 013226  013737  001346  001140          MOV     RMCS2I,$GDDAT           ;EXPECTED STATUS
    464 013234  052737  040000  001140          BIS     #WCE,$GDDAT
    465 013242  013737  001346  001142          MOV     RMCS2I,$BDDAT           ;RECEIVED STATUS
    466 013250  010337  001174                  MOV     R3,$TMP0                ;FAILING BIT POSITION
    467 013254  012737  107440  001176          MOV     #BUFTWO-2,$TMP1         ;FAILING ADDRESS
    468 013262  104337                          EMT     337
    469 013264  000470                          BR      29$
    470 013266                          24$:
    471 013266  112737  000022  001524          MOVB    #RMDB,GETINX    ;SETUP GET INDEX TABLE
        013274  112737  000200  001525          MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
        013302  004737  043060                  JSR     PC,GET          ;GO READ RMDB VIA GET SUBROUTINE
        013306  000404                          BR      25$             ;GO TO 25$ IF NO ERROR
        013310  000240                          NOP                     ;RETURN HERE IF ERROR
        013312  104000                          EMT                     ;ERROR DEFINED BY GET SUBROUTINE
        013314  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR
    472 013320                          25$:
    473 013320  013737  001360  001142          MOV     RMDBI,$BDDAT            ;RECEIVED DATA
    474 013326  013737  107440  001140          MOV     BUFTWO-2,$GDDAT         ;EXPECTED DATA
    475 013334  040337  001140                  BIC     R3,$GDDAT
    476 013340  012737  107440  001134          MOV     #BUFTWO-2,$GDADR        ;EXPECTED ADDRESS
    477 013346  013737  001342  001136          MOV     RMBAI,$BDADR            ;RECEIVED ADDRESS
    478 013354  162737  000002  001136          SUB     #2,$BDADR               ;ADJUST BUS ADDRESS
    479 013362  023737  001134  001136          CMP     $GDADR,$BDADR           ;ADDRESSES OK??
    480 013370  001402                          BEQ     26$                     ;YES.!
    481 013372  104340                          EMT     340
    482 013374  000424                          BR      29$
    483 013376  023737  001140  001142  26$:    CMP     $GDDAT,$BDDAT           ;DATA OK??
    484 013404  001402                          BEQ     27$                     ;YES.!
    485 013406  104341                          EMT     341
    486 013410  000416                          BR      29$
    487 013412                          27$:
    488
    489 013412  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        013416  000405                          BR      28$             ;GO TO 28$ IF NO ERROR
        013420  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        013422  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        013424  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013426  000137  013446                  JMP     29$             ;GO TO 29$ IF ERROR
    490 013432                          28$:
    491 013432  040337  107440                  BIC     R3,BUFTWO-2             ;RESTORE DATA PATTERN
    492 013436  006303                          ASL     R3                      ;SHIFT TO NEXT BIT
    493 013440  001402                          BEQ     29$                     ;EXIT IF DONE
    494 013442  000137  013020                  JMP     17$                     ;REPEAT TEST FOR NEXT DATA BIT
    495 013446                          29$:
    496
    497                                 ;;*******************************************************************
                                        ;*TEST 5         WRITE, READ ONES
                                        ;;*******************************************************************
        013446                          TST5:
        013446  000004                          SCOPE                   ;SCOPE CALL
        013450  000240                          NOP                     ;START OF TEST
        013452  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        013456  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        013462  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        013466  012737  000005  001226          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
    498
    499                                 ;;*******************************************************************
    500                                 ;LOOP #1        FORMAT,WRITE,READ
    501
    502 013474                          1$:
    503
    504                                 ;PREPARE THE DEVICE FOR TEST
    505 013474  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        013500  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        013502  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
        013504  000240                          NOP                     ;RETURN HERE IF ERROR
        013506  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        013510  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    506
    507                                 ;LOAD PARAMETERS AND GENERATE DATA BUFFER
    508 013514                          2$:
    509 013514  012737  000000  001446          MOV     #0,RMDCO                ;CYLINDER = 0
    510 013522  012737  000000  001420          MOV     #0,RMDAO                ;TRACK = 0, SECTOR   0
    511 013530  012737  106436  001416          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
    512 013536  012737  177376  001414          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
    513 013544  012737  010000  001444          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
    514 013552  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
    515
    516                                 ;VERIFY THAT SECTOR IS NOT BAD
        013560  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        013564  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        013566  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        013572  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        013574  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    517 013600                          3$:
```

```
                                                ;STARTING ADDRESS OF PATTERN
                                                ;RANGE OF PATTERN
                                                ;GO GENERATE BUFFER FOR FORMAT

                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
                                                ;R2 = BYTE ENTRY POSITION
                                                ;TERMINATE TABLE
                        4$:

                        ;FORMAT THE DRIVE
                                ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                ;GO TO 5$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY PUT SUBROUTINE
                                ;GO TO 25$ IF ERROR
                        5$:

                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
                                ;GO TO GETSTS SUBROUTINE

                        ;WAIT FOR COMMAND TO COMPLETE
                                ;GO TO TIMOUT SUBROUTINE

                        ;GO READ STATUS FOR FORMAT OPERATION
                                ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                ;GO TO 6$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY GET SUBROUTINE
                                ;GO TO 25$ IF ERROR
                        6$:

                        ;VERIFY NO ERRORS DURING FORMAT
                                ;GO VERIFY RESULTS OF DATA TRANSFER
                                ;GO TO 7$ IF NO ERROR
                                ;RETURN HERE IF ERROR
                                ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                ;GO BACK FOR MORE ERROR CHECKS
                                ;GO TO 25$ IF ERROR
                        7$:

                        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
                                                ;SKIP TO WRITE OPERATION

                        ;;*****************************************************************
                        ;LOOP #2        WRITE,READ

                        8$:

                        ;PREPARE DEVICE FOR TEST
```

```
                                                                ;PREPARE DEVICE FOR TEST
560 013754  004737  037316          JSR     PC,TSTPRP           ;TASK DESCRIPTOR AS FOLLOWS:
    013760  154130                  .WORD   154130              ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
    013762  000404                  BR      9$                  ;GO TO 9$ IF NO ERROR
    013764  000240                  NOP                         ;RETURN HERE IF ERROR
    013766  104000                  EMT                         ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    013770  000137  014454          JMP     25$                 ;GO TO 25$ IF ERROR
561 013774                      9$:
562
563                             ;WRITE DATA TO THE DRIVE
564 013774                      10$:
565 013774  012737  106442  001416      MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
566 014002  012737  177400  001414      MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
567 014010  012737  000061  001412      MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
568 014016  012702  001553              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
569 014022  112722  000006              MOVB    #RMDA,(R2)+
570 014026  112722  000034              MOVB    #RMDC,(R2)+
571 014032  112722  000032              MOVB    #RMOF,(R2)+
572 014036  112722  000004              MOVB    #RMBA,(R2)+
573 014042  112722  000002              MOVB    #RMWC,(R2)+
574 014046  112722  000000              MOVB    #RMCS1,(R2)+
575 014052  112722  000200              MOVB    #200,(R2)+              ;TERMINATE TABLE
576
577 014056  004737  043330          JSR     PC,PUT              ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    014062  000404                  BR      11$                 ;GO TO 11$ IF NO ERROR
    014064  000240                  NOP                         ;RETURN HERE IF ERROR
    014066  104000                  EMT                         ;ERROR # DEFINED BY PUT SUBROUTINE
    014070  000137  014454          JMP     25$                 ;GO TO 25$ IF ERROR
578 014074                      11$:
579
580                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    014074  004737  042774          JSR     PC,GETSTS           ;GO TO GETSTS SUBROUTINE
581
582                             ;WAIT FOR COMMAND TO COMPLETE
    014100  004737  043672          JSR     PC,TIMOUT           ;GO TO TIMOUT SUBROUTINE
583
584                             ;GO READ STATUS FOR WRITE COMMAND
585 014104  004737  043060          JSR     PC,GET              ;GO READ REGISTER(S) WITH GET SUBROUTINE
    014110  000404                  BR      12$                 ;GO TO 12$ IF NO ERROR
    014112  000240                  NOP                         ;RETURN HERE IF ERROR
    014114  104000                  EMT                         ;ERROR # DEFINED BY GET SUBROUTINE
    014116  000137  014454          JMP     25$                 ;GO TO 25$ IF ERROR
586 014122                      12$:
587
588                             ;CHECK FOR ERRORS DURING WRITE OPERATION
589 014122  004737  044056          JSR     PC,PRIERR           ;GO CHECK FOR PRIMARY ERRORS
    014126  000405                  BR      13$                 ;GO TO 13$ IF NO ERROR
    014130  000240                  NOP                         ;RETURN HERE IF ERROR
    014132  104000                  EMT                         ;ERROR # DEFINED BY PRIERR SUBROUTINE
    014134  004736                  JSR     PC,@(SP)+           ;GO BACK FOR MORE ERROR CHECKS
    014136  000137  014454          JMP     25$                 ;GO TO 25$ IF ERROR
```

```
                                        13$:
014142  004737  056572                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
014146  000405                                BR      14$             ;GO TO 14$ IF NO ERROR
014150  000240                                NOP                     ;RETURN HERE IF ERROR
014152  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
014154  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014156  000137  014454                        JMP     25$             ;GO TO 25$ IF ERROR
014162                                  14$:
014162  004737  044710                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
014166  000405                                BR      15$             ;GO TO 15$ IF NO ERROR
014170  000240                                NOP                     ;RETURN HERE IF ERROR
014172  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
014174  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014176  000137  014454                        JMP     25$             ;GO TO 25$ IF ERROR
014202                                  15$:

                                        ;CHANGE LOOP ADDRESSES
014202  012737  014212  001124                MOV     #16$,$LPERR
014210  000410                                BR      17$                     ;SKIP TO NEXT OPERATION

                                        ;;*****************************************************************
                                        ;LOOP #3        READ

014212                                  16$:

                                        ;PREPARE DEVICE FOR TEST
014212  004737  037316                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
014216  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "P1P" IS SET
                                                                        ;VERIFY RECALIBRATION
014220  000404                                BR      17$             ;GO TO 17$ IF NO ERROR
014222  000240                                NOP                     ;RETURN HERE IF ERROR
014224  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
014226  000137  014454                        JMP     25$             ;GO TO 25$ IF ERROR
014232                                  17$:

                                        ;READ DATA FROM DEVICE
014232                                  18$:
014232  012737  107446  001416                MOV     #BUFTWO+4,RMBAO         ;CHANGE MEMORY ADDRESS
014240  012737  000071  001412                MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
014246  012702  001553                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
014252  112722  000006                        MOVB    #RMDA,(R2)+
014256  112722  000032                        MOVB    #RMOF,(R2)+
014262  112722  000034                        MOVB    #RMDC,(R2)+
014266  112722  000004                        MOVB    #RMBA,(R2)+
014272  112722  000002                        MOVB    #RMWC,(R2)+
014276  112722  000000                        MOVB    #RMCS1,(R2)+
014302  112712  000200                        MOVB    #200,(R2)

014306  004737  043330                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
014312  000404                                BR      19$             ;GO TO 19$ IF NO ERROR
014314  000240                                NOP                     ;RETURN HERE IF ERROR
```

```
        014316  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        014320  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    623 014324                          19$:
    624
    625                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        014324  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    626
    627                                 ;WAIT FOR COMMAND TO COMPLETE
        014330  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    628
    629                                 ;GO READ STATUS FOR READ OPERATION
    630 014334  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        014340  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        014342  000240                          NOP                     ;RETURN HERE IF ERROR
        014344  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014346  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    631 014352                          20$:
    632
    633                                 ;CHECK FOR ERRORS DURING READ OPERATION
    634 014352  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        014356  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
        014360  000240                          NOP                     ;RETURN HERE IF ERROR
        014362  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        014364  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014366  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    635 014372                          21$:
    636 014372  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        014376  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
        014400  000240                          NOP                     ;RETURN HERE IF ERROR
        014402  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        014404  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014406  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    637 014412                          22$:
    638 014412  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        014416  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
        014420  000240                          NOP                     ;RETURN HERE IF ERROR
        014422  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        014424  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014426  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    639 014432                          23$:
    640 014432  004737  042432                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        014436  106442                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        014440  107446                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        014442  000404                          BR      24$             ;GO TO 24$ IF NO ERROR
        014444  000240                          NOP                     ;RETURN HERE IF ERROR
        014446  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        014450  000137  014454                  JMP     25$             ;GO TO 25$ IF ERROR
    641 014454                          24$:
    642
    643 014454                          25$:
    644
    645
                                        ;;*******************************************************************
                                        ;*TEST 6         WRITE, WRITE CHECK ONES
                                        ;;*******************************************************************
        014454                          TST6:
        014454  000004                          SCOPE                   ;SCOPE CALL
        014456  000240                          NOP                     ;START OF TEST
```

```
        014460  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        014464  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        014470  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        014474  012737  000006  001226          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
    646
    647                                 ;;**************************************************************
    648                                 ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
    649
    650 014502                          1$:
    651
    652                                 ;PREPARE THE DEVICE FOR TEST
    653 014502  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        014506  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SFT
                                                                        ;VERIFY RECALIBRATION
        014510  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
        014512  000240                          NOP                     ;RETURN HERE IF ERROR
        014514  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        014516  000137  015432                  JMP     24$             ;GO TO 24$ IF ERROR
    654
    655                                 ;LOAD PARAMETERS AND GENERATE DATA BUFFER
    656 014522                          2$:
    657 014522  012737  000000  001446          MOV     #0,RMDC0                ;CYLINDER = 0
    658 014530  012737  000000  001420          MOV     #0,RMDA0                ;TRACK = 0, SECTOR = 0
    659 014536  012737  106436  001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
    660 014544  012737  177376  001414          MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
    661 014552  012737  010000  001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
    662 014560  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
    663
    664                                 ;VERIFY THAT SECTOR IS NOT BAD
        014566  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        014572  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        014574  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        014600  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        014602  000137  015432                  JMP     24$             ;GO TO 24$ IF ERROR
    665 014606                          3$:
    666 014606  012737  072016  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
    667 014614  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
    668 014622  004737  042174                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
    669
    670                                 ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
    671 014626  012702  001553                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
    672 014632  112722  000034                  MOVB    #RMDC,(R2)+
    673 014636  112722  000006                  MOVB    #RMDA,(R2)+
    674 014642  112722  000004                  MOVB    #RMBA,(R2)+
    675 014646  112722  000002                  MOVB    #RMWC,(R2)+
    676 014652  112722  000032                  MOVB    #RMOF,(R2)+
    677 014656  112722  000000                  MOVB    #RMCS1,(R2)+
    678 014662  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
    679 014666                          4$:
    680
    681                                 ;FORMAT THE DRIVE
```

```
   682 014666  004737  043330                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
       014672  000404                            BR      5$              ;GO TO 5$ IF NO ERROR
       014674  000240                            NOP                     ;RETURN HERE IF ERROR
       014676  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
       014700  000137  015432                    JMP     24$             ;GO TO 24$ IF ERROR
   683 014704                              5$:
   684
   685                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
       014704  004737  042774                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
   686
   687                                 ;WAIT FOR COMMAND TO COMPLETE
       014710  004737  043672                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
   688
   689                                 ;GO READ STATUS FOR FORMAT OPERATION
   690 014714  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
       014720  000404                            BR      6$              ;GO TO 6$ IF NO ERROR
       014722  000240                            NOP                     ;RETURN HERE IF ERROR
       014724  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
       014726  000137  015432                    JMP     24$             ;GO TO 24$ IF ERROR
   691 014732                              6$:
   692
   693                                 ;VERIFY NO ERRORS DURING FORMAT
   694 014732  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
       014736  000405                            BR      7$              ;GO TO 7$ IF NO ERROR
       014740  000240                            NOP                     ;RETURN HERE IF ERROR
       014742  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
       014744  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
       014746  000137  015432                    JMP     24$             ;GO TO 24$ IF EPROR
   695 014752                              7$:
   696
   697                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
   698 014752  012737  014762  001124            MOV     #8$,$LPERR
   699 014760  000410                            BR      9$                      ;SKIP TO WRITE OPERATION
   700
   701                                 ;;*************************************************************************
   702                                 ;LOOP #2        WRITE,WRITE CHECK DATA
   703
   704 014762                              8$:
   705
   706
   707                                 ;PREPARE DEVICE FOR WRITE OPERATION
   708 014762  004737  037316                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
       014766  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                          ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                          ;CLEAR CONTROLLER & SELECT DEVICE
                                                                          ;VERIFY CONTROLLER CLEAR OPERATION
                                                                          ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                          ;VERIFY PACK ACKNOWLEDGE
                                                                          ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                          ;VERIFY RECALIBRATION
       014770  000404                            BR      9$              ;GO TO 9$ IF NO ERROR
       014772  000240                            NOP                     ;RETURN HERE IF ERROR
       014774  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
       014776  000137  015432                    JMP     24$             ;GO TO 24$ IF ERROR
   709 015002                              9$:
   710
   711                                 ;WRITE DATA TO THE DRIVE
```

```
                                        10$:
015002  012737  106442  001416                  MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
015010  012737  177400  001414                  MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
015016  012737  000061  001412                  MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
015024  012702  001553                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
015030  112722  000006                          MOVB    #RMDA,(R2)+
015034  112722  000034                          MOVB    #RMDC,(R2)+
015040  112722  000032                          MOVB    #RMOF,(R2)+
015044  112722  000004                          MOVB    #RMBA,(R2)+
015050  112722  000002                          MOVB    #RMWC,(R2)+
015054  112722  000000                          MOVB    #RMCS1,(R2)+
015060  112722  000200                          MOVB    #200,(R2)+              ;TERMINATE TABLE

015064  004737  043330                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
015070  000404                                  BR      11$             ;GO TO 11$ IF NO ERROR
015072  000240                                  NOP                     ;RETURN HERE IF ERROR
015074  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
015076  000137  015432                          JMP     24$             ;GO TO 24$ IF ERROR
015102                                  11$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
015102  004737  042774                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
015106  004737  043672                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE COMMAND
015112  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
015116  000404                                  BR      12$             ;GO TO 12$ IF NO ERROR
015120  000240                                  NOP                     ;RETURN HERE IF ERROR
015122  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
015124  000137  015432                          JMP     24$             ;GO TO 24$ IF ERROR
015130                                  12$:

                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
015130  004737  044056                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
015134  000405                                  BR      13$             ;GO TO 13$ IF NO ERROR
015136  000240                                  NOP                     ;RETURN HERE IF ERROR
015140  104000                                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
015142  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015144  000137  015432                          JMP     24$             ;GO TO 24$ IF ERROR
015150                                  13$:
015150  004737  056572                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
015154  000405                                  BR      14$             ;GO TO 14$ IF NO ERROR
015156  000240                                  NOP                     ;RETURN HERE IF ERROR
015160  104000                                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
015162  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015164  000137  015432                          JMP     24$             ;GO TO 24$ IF ERROR
015170                                  14$:
015170  004737  044710                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
015174  000405                                  BR      15$             ;GO TO 15$ IF NO ERROR
015176  000240                                  NOP                     ;RETURN HERE IF ERROR
015200  104000                                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
015202  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015204  000137  015432                          JMP     24$             ;GO TO 24$ IF ERROR
015210                                  15$:
```

```
                                        ;CHANGE LOOP ADDRESSES
015210  012737  015220  001124                MOV     #16$,$LPERR
015216  000410                                BR      17$                     ;SKIP TO NEXT OPERATION

                                        ;;*****************************************************************
                                        ;LOOP #3        WRITE CHECK DATA

015220                                  16$:

                                        ;PREPARE DEVICE FOR WRITE CHECK DATA OPERATION
015220  004737  037316                        JSR     PC,TSTPRP               ;PREPARE DEVICE FOR TEST
015224  154130                                .WORD   154130                  ;TASK DESCRIPTOR AS FOLLOWS:
                                                                            ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                            ;VERIFY PACK ACKNOWLEDGE
                                                                            ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                            ;VERIFY RECALIBRATION
015226  000404                                BR      17$                     ;GO TO 17$ IF NO ERROR
015230  000240                                NOP                             ;RETURN HERE IF ERROR
015232  104000                                EMT                             ;ERROR # DEFINED BY TSTPRP SUBROUTINE
015234  000137  015432                        JMP     24$                     ;GO TO 24$ IF ERROR
015240                                  17$:

                                        ;WRITE CHECK DATA DATA FROM DEVICE
015240                                  18$:
015240  012737  000051  001412                MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
015246  012702  001553                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
015252  112722  000006                        MOVB    #RMDA,(R2)+
015256  112722  000032                        MOVB    #RMOF,(R2)+
015262  112722  000034                        MOVB    #RMDC,(R2)+
015266  112722  000004                        MOVB    #RMBA,(R2)+
015272  112722  000002                        MOVB    #RMWC,(R2)+
015276  112722  000000                        MOVB    #RMCS1,(R2)+
015302  112712  000200                        MOVB    #200,(R2)

015306  004737  043330                        JSR     PC,PUT                  ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
015312  000404                                BR      19$                     ;GO TO 19$ IF NO ERROR
015314  000240                                NOP                             ;RETURN HERE IF ERROF
015316  104000                                EMT                             ;ERROR # DEFINED BY PUT SUBROUTINE
015320  000137  015432                        JMP     24$                     ;GO TO 24$ IF ERROR
015324                                  19$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
015324  004737  042774                        JSR     PC,GETSTS               ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
015330  004737  043672                        JSR     PC,TIMOUT               ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
015334  004737  043060                        JSR     PC,GET                  ;GO READ REGISTER(S) WITH GET SUBROUTINE
015340  000404                                BR      20$                     ;GO TO 20$ IF NO ERROR
015342  000240                                NOP                             ;RETURN HERE IF ERROR
015344  104000                                EMT                             ;ERROR # DEFINED BY GET SUBROUTINE
015346  000137  015432                        JMP     24$                     ;GO TO 24$ IF ERROR
015352                                  20$:
```

```
                                        ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
015352  004737  044056                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
015356  000405                                BR      21$             ;GO TO 21$ IF NO ERROR
015360  000240                                NOP                     ;RETURN HERE IF ERROR
015362  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
015364  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015366  000137  015432                        JMP     24$             ;GO TO 24$ IF ERROR
015372                                  21$:
015372  004737  056572                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
015376  000405                                BR      22$             ;GO TO 22$ IF NO ERROR
015400  000240                                NOP                     ;RETURN HERE IF ERROR
015402  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
015404  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015406  000137  015432                        JMP     24$             ;GO TO 24$ IF ERROR
015412                                  22$:
015412  004737  044710                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
015416  000405                                BR      23$             ;GO TO 23$ IF NO ERROR
015420  000240                                NOP                     ;RETURN HERE IF ERROR
015422  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
015424  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015426  000137  015432                        JMP     24$             ;GO TO 24$ IF ERROR
015432                                  23$:

015432                                  24$:

                                        ;;*******************************************************************
                                        ;*TEST 7        WRITE, WRITE CHECK ONES W/ WCE ERROR
                                        ;;*******************************************************************
015432                                  TST7:
015432  000004                                SCOPE                   ;SCOPE CALL
015434  000240                                NOP                     ;START OF TEST
015436  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
015442  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
015446  013701  001466                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
015452  012737  000007  001226                MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

                                        ;;*******************************************************************
                                        ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA

015460                                  1$:

                                        ;PREPARE THE DEVICE FOR FORMAT OPERATION
015460  004737  037316                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
015464  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
015466  000404                                BR      2$              ;GO TO 2$ IF NO ERROR
015470  000240                                NOP                     ;RETURN HERE IF ERROR
015472  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
015474  000137  016630                        JMP     29$             ;GO TO 29$ IF ERROR
015500                                  2$:
```

```
800
801                                     ;LOAD PARAMETERS AND GENERATE DATA BUFFER
802 015500  012737  000000  001446              MOV     #0,RMDCO                ;CYLINDER = 0
803 015506  012737  000000  001420              MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
804 015514  012737  106436  001416              MOV     #BUFONE,RMBAO           ;BUS ADDRESS
805 015522  012737  177376  001414              MOV     #-258.,RMWCO            ;WORD COUNT = 1 SECTOR
806 015530  012737  010000  001444              MOV     #FMT16,RMOFO            ;16 BIT FORMAT
807 015536  012737  000063  001412              MOV     #WHIGO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
808
809                                     ;VERIFY THAT SECTOR IS NOT BAD
    015544  004737  040242                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    015550  000405                              BR      3$              ;GO TO 3$ IF NO ERROR
    015552  104401  070426                      TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    015556  104000                              EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    015560  000137  016630                      JMP     29$             ;GO TO 29$ IF ERROR
810 015564                              3$:
811 015564  012737  072016  001174              MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
812 015572  012737  000001  001176              MOV     #1,$TMP1                ;RANGE OF PATTERN
813 015600  004737  042174                      JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
814
815                                     ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
816 015604  012702  001553                      MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
817 015610  112722  000034                      MOVB    #RMDC,(R2)+
818 015614  112722  000006                      MOVB    #RMDA,(R2)+
819 015620  112722  000004                      MOVB    #RMBA,(R2)+
820 015624  112722  000002                      MOVB    #RMWC,(R2)+
821 015630  112722  000032                      MOVB    #RMOF,(R2)+
822 015634  112722  000000                      MOVB    #RMCS1,(R2)+
823 015640  112712  000200                      MOVB    #200,(R2)               ;TERMINATE TABLE
824 015644                              4$:
825
826                                     ;FORMAT THE DRIVE
827 015644  004737  043330                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    015650  000404                              BR      5$              ;GO TO 5$ IF NO ERROR
    015652  000240                              NOP                     ;RETURN HERE IF ERROR
    015654  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    015656  000137  016630                      JMP     29$             ;GO TO 29$ IF ERROR
828 015662                              5$:
829
830                                     ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    015662  004737  042774                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
831
832                                     ;WAIT FOR COMMAND TO COMPLETE
    015666  004737  043672                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
833
834                                     ;GO READ STATUS FOR FORMAT OPERATION
835 015672  004737  043060                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    015676  000404                              BR      6$              ;GO TO 6$ IF NO ERROR
    015700  000240                              NOP                     ;RETURN HERE IF ERROR
    015702  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
    015704  000137  016630                      JMP     29$             ;GO TO 29$ IF ERROR
836 015710                              6$:
837
838                                     ;VERIFY NO ERRORS DURING FORMAT
839 015710  004737  056572                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
    015714  000405                              BR      7$              ;GO TO 7$ IF NO ERROR
    015716  000240                              NOP                     ;RETURN HERE IF ERROR
```

```
        015720  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        015722  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015724  000137  016630                  JMP     29$             ;GO TO 29$ IF ERROR
   840  015730                          7$:
   841
   842                                  ;MOVE LOOP ADDRESSES TO NEXT OPERATION
   843  015730  012737  015740  001124          MOV     #8$,$LPERR
   844  015736  000410                          BR      9$                      ,SKIP TO WRITE OPERATION
   845
   846                                  ;;*****************************************************************
   847                                  ;LOOP #2        WRITE,WRITE CHECK DATA
   848
   849  015740                          8$:
   850
   851                                  ;PREPARE DEVICE FOR WRITE OPERATION
   852  015740  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        015744  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        015746  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
        015750  000240                          NOP                     ;RETURN HERE IF ERROR
        015752  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        015754  000137  016630                  JMP     29$             ;GO TO 29$ IF ERROR
   853  015760                          9$:
   854
   855                                  ;WRITE DATA TO THE DRIVE
   856  015760                          10$:
   857  015760  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
   858  015766  012737  106442  001416          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
   859  015774  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
   860  016002  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
   861  016006  112722  000006                  MOVB    #RMDA,(R2)+
   862  016012  112722  000034                  MOVB    #RMDC,(R2)+
   863  016016  112722  000032                  MOVB    #RMOF,(R2)+
   864  016022  112722  000004                  MOVB    #RMBA,(R2)+
   865  016026  112722  000002                  MOVB    #RMWC,(R2)+
   866  016032  112722  000000                  MOVB    #RMCS1,(R2)+
   867  016036  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
   868
   869  016042  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        016046  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
        016050  000240                          NOP                     ;RETURN HERE IF ERROR
        016052  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        016054  000137  016630                  JMP     29$             ;GO TO 29$ IF ERROR
   870  016060                          11$:
   871
   872                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        016060  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
   873
   874                                  ;WAIT FOR COMMAND TO COMPLETE
        016064  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
   875
```

```
                                        ;GO READ STATUS FOR WRITE COMMAND
016070  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016074  000404                                  BR      12$             ;GO TO 12$ IF NO ERROR
016076  000240                                  NOP                     ;RETURN HERE IF ERROR
016100  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
016102  000137  016630                          JMP     29$             ;GO TO 29$ IF ERROR
016106                                  12$:

                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
016106  004737  044056                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
016112  000405                                  BR      13$             ;GO TO 13$ IF NO ERROR
016114  000240                                  NOP                     ;RETURN HERE IF ERROR
016116  104000                                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
016120  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016122  000137  016630                          JMP     29$             ;GO TO 29$ IF ERROR
016126                                  13$:
016126  004737  056572                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
016132  000405                                  BR      14$             ;GO TO 14$ IF NO ERROR
016134  000240                                  NOP                     ;RETURN HERE IF ERROR
016136  104000                                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
016140  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016142  000137  016630                          JMP     29$             ;GO TO 29$ IF ERROR
016146                                  14$:
016146  004737  044710                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
016152  000405                                  BR      15$             ;GO TO 15$ IF NO ERROR
016154  000240                                  NOP                     ;RETURN HERE IF ERROR
016156  104000                                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
016160  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016162  000137  016630                          JMP     29$             ;GO TO 29$ IF ERROR
016166                                  15$:

                                        ;CHANGE LOOP ADDRESSES
016166  012737  016174  001124                  MOV     #16$,$LPERR

                                        ;;*****************************************************************
                                        ;LOOP #3        WRITE CHECK DATA

016174                                  16$:

016174  012703  000001                          MOV     #1,R3                   ;R3=WCE BIT POSITION
016200  040337  107440                  17$:    BIC     R3,BUFTWO-2             ;CHANGE LAST WORD OF BUFFER

                                        ;PREPARE DEVICE FOR WRITE CHECK DATA OPERATION
016204  004737  037316                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
016210  154130                                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
016212  000404                                  BR      18$             ;GO TO 18$ IF NO ERROR
016214  000240                                  NOP                     ;RETURN HERE IF ERROR
016216  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
016220  000137  016630                          JMP     29$             ;GO TO 29$ IF ERROR
016224                                  18$:
```

```
                                        ;READ DATA FROM DEVICE
016224                                  19$:
016224  012737  000051  001412                MOV     #WCD!GO,RMCS1O          ;WRITE CHECK DATA DATA COMMAND
016232  012702  001553                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
016236  112722  000006                        MOVB    #RMDA,(R2)+
016242  112722  000032                        MOVB    #RMOF,(R2)+
016246  112722  000034                        MOVB    #RMDC,(R2)+
016252  112722  000004                        MOVB    #RMBA,(R2)+
016256  112722  000002                        MOVB    #RMWC,(R2)+
016262  112722  000000                        MOVB    #RMCS1,(R2)+
016266  112712  000200                        MOVB    #200,(R2)

016272  004737  043330                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
016276  000404                                BR      20$             ;GO TO 20$ IF NO ERROR
016300  000240                                NOP                     ;RETURN HERE IF ERROR
016302  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
016304  000137  016630                        JMP     29$             ;GO TO 29$ IF ERROR
016310                                  20$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
016310  004737  042774                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
016314  004737  043672                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
016320  004737  043060                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016324  000404                                BR      21$             ;GO TO 21$ IF NO ERROR
016326  000240                                NOP                     ;RETURN HERE IF ERROR
016330  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
016332  000137  016630                        JMP     29$             ;GO TO 29$ IF ERROR
016336                                  21$:

                                        ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
016336  004737  044056                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
016342  000405                                BR      22$             ;GO TO 22$ IF NO ERROR
016344  000240                                NOP                     ;RETURN HERE IF ERROR
016346  104000                                EMT                     ;ERPOR # DEFINED BY PRIERR SUBROUTINE
016350  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016352  000137  016630                        JMP     29$             ;GO TO 29$ IF ERROR
016356                                  22$:
016356  032737  040000  001346                BIT     #WCE,RMCS2I             ;WAS 'WCE' DETECTED??
016364  001030                                BNE     24$                     ;YES!!
016366  004737  056572                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
016372  000405                                BR      23$             ;GO TO 23$ IF NO ERROR
016374  000240                                NOP                     ;RETURN HERE IF ERROR
016376  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
016400  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016402  000137  016630                        JMP     29$             ;GO TO 29$ IF ERROR
016406                                  23$:
016406  013737  001346  001140                MOV     RMCS2I,$GDDAT           ;EXPECTED STATUS
016414  052737  040000  001140                BIS     #WCE,$GDDAT
016422  013737  001346  001142                MOV     RMCS2I,$BDDAT           ;RECEIVED STATUS
016430  010337  001174                        MOV     R3,$TMP0                ;FAILING BIT POSITION
016434  012737  107440  001176                MOV     #BUFTWO-2,$TMP1         ;FAILING ADDRESS
016442  104337                                EMT     337
```

```
939 016444  000471                          BR      29$
940
941 016446                          24$:
942 016446  112737  000022  001524          MOVB    #RMDB,GETINX    ;SETUP GET INDEX TABLE
    016454  112737  000200  001525          MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
    016462  012737  016630  001360          MOV     #29$,RMDB+      ;SET RMDB INPUT BUFFER = 29$
    016470  004737  043060                  JSR     PC,GET          ;GO READ RMDB VIA GET SUBROUTINE
    016474  000402                          BR      25$             ;GO TO 25$ IF NO ERROR
    016476  000240                          NOP                     ;RETURN HERE IF ERROR
    016500  104000                          EMT                     ;ERROR DEFINED BY GET SUBROUTINE
943
944 016502                          25$:
945 016502  013737  001360  001142          MOV     RMDBI,$BDDAT            ;RECEIVED DATA
946 016510  013737  107440  001140          MOV     BUFTWO-2,$GDDAT         ;EXPECTED DATA
947 016516  050337  001140                  BIS     R3,$GDDAT
948 016522  012737  107440  001134          MOV     #BUFTWO-2,$GDADR        ;EXPECTED ADDRESS
949 016530  013737  001342  001136          MOV     RMBAI,$BDADR            ;RECEIVED ADDRESS
950 016536  162737  000002  001136          SUB     #2,$BDADR               ;CORRECT MEMORY ADDRESS
951 016544  023737  001134  001136          CMP     $GDADR,$BDADR           ;ADDRESSES OK??
952 016552  001402                          BEQ     26$                     ;YES!!
953 016554  104340                          EMT     340
954 016556  000424                          BR      29$
955 016560  023737  001140  001142  26$:    CMP     $GDDAT,$BDDAT           ;DATA OK??
956 016566  001402                          BEQ     27$                     ;YES!.
957 016570  104341                          EMT     341
958 016572  000416                          BR      29$
959 016574                          27$:
960
961 016574  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
    016600  000405                          BR      28$             ;GO TO 28$ IF NO ERROR
    016602  000240                          NOP                     ;RETURN HERE IF ERROR
    016604  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
    016606  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    016610  000137  016630                  JMP     29$             ;GO TO 29$ IF ERROR
962 016614                          28$:
963 016614  050337  107440                  BIS     R3,BUFTWO-2             ;RESTORE DATA PATTERN
964 016620  006303                          ASL     R3                      ;SHIFT TO NEXT BIT
965 016622  001402                          BEQ     29$                     ;EXIT IF DONE
966 016624  000137  016200                  JMP     17$                     ;REPEAT TEST FOR NEXT DATA BIT
967 016630                          29$:
968
969                                 ;;*****************************************************************
                                    ;*TEST 10        WRITE, WRITE CHECK MULTIPLE SECTORS
                                    ;;*****************************************************************
    016630                          TST10:
    016630  000004                          SCOPE                   ;SCOPE CALL
    016632  000240                          NOP                     ;START OF TEST
    016634  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
    016640  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    016644  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
    016650  012737  000010  001226          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
970
971                                 ;;*****************************************************************
972                                  LOOP #1        FORMAT,WRITE,WRITE CHECK
973
974 016656                          1$:
975
```

```
                                        ;PREPARE THE DEVICE FOR FORMAT OPERATION
016656  004737  037316                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
016662  154130                                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
016664  000404                                  BR      2$              ;GO TO 2$ IF NO ERROR
016666  000240                                  NOP                     ;RETURN HERE IF ERROR
016670  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
016672  000137  017634                          JMP     24$             ;GO TO 24$ IF ERROR
016676                                  2$:

                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
016676  012737  000000  001446                  MOV     #0,RMDC0                ;CYLINDER = 0
016704  012737  000000  001420                  MOV     #0,RMDA0                ;TRACK = 0, SECTOR = 0
016712  012737  106436  001416                  MOV     #BUFONE,RMBA0           ;BUS ADDRESS
016720  012737  176774  001414                  MOV     #-258.*2,RMWC0          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
016726  012737  010000  001444                  MOV     #FMT16,RMOF0            ;16 BIT FORMAT
016734  012737  000063  001412                  MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND

                                        ;VERIFY THAT SECTOR IS NOT BAD
016742  004737  040242                          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
016746  000405                                  BR      3$              ;GO TO 3$ IF NO ERROR
016750  104401  070426                          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
016754  104000                                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
016756  000137  017634                          JMP     24$             ;GO TO 24$ IF ERROR
016762                                  3$:
016762  012737  072060  001174                  MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
016770  012737  000001  001176                  MOV     #1,$TMP1                ;RANGE OF PATTERN
016776  004737  042174                          JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
017002  012702  001553                          MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
017006  112722  000034                          MOVB    #RMDC,(R2)+
017012  112722  000006                          MOVB    #RMDA,(R2)+
017016  112722  000004                          MOVB    #RMBA,(R2)+
017022  112722  000002                          MOVB    #RMWC,(R2)+
017026  112722  000032                          MOVB    #RMOF,(R2)+
017032  112722  000000                          MOVB    #RMCS1,(R2)+
017036  112712  000200                          MOVB    #200 (R2)               ;TERMINATE TABLE
017042                                  4$:

                                        ;FORMAT THE DRIVE
017042  004737  043330                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
017046  000404                                  BR      5$              ;GO TO 5$ IF NO ERROR
017050  000240                                  NOP                     ;RETURN HERE IF ERROR
017052  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
017054  000137  017634                          JMP     24$             ;GO TO 24$ IF ERROR
017060                                  5$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
017060  004737  042774                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
```

```
1011                                    ;WAIT FOR COMMAND TO COMPLETE
     017064  004737  043672                     JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1012
1013                                    ;GO READ STATUS FOR FORMAT OPERATION
1014 017070  004737  043060                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     017074  000404                            BR      6$              ;GO TO 6$ IF NO ERROR
     017076  000240                            NOP                     ;RETURN HERE IF ERROR
     017100  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     017102  000137  017634                     JMP     24$             ;GO TO 24$ IF ERROR
1015 017106                             6$:
1016
1017                                    ;VERIFY NO ERRORS DURING FORMAT
1018 017106  004737  056572                     JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     017112  000405                            BR      7$              ;GO TO 7$ IF NO ERROR
     017114  000240                            NOP                     ;RETURN HERE IF ERROR
     017116  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     017120  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017122  000137  017634                     JMP     24$             ;GO TO 24$ IF ERROR
1019 017126                             7$:
1020
1021                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1022 017126  012737  017200  001124             MOV     #8$,$LPERR
1023
1024                                    ;REGENERATE DATA BUFFER
1025 017134  012737  177000  001414             MOV     #-256.*2,RMWCO  ;CHANGE WORD COUNT
1026 017142  012737  106436  001416             MOV     #BUFONE,RMBAO           ;CHANGE BUS ADDRESS
1027 017150  012737  000060  001412             MOV     #WD,RMCS1O              ;WRITE DATA
1028 017156  012737  072016  001174             MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
1029 017164  012737  000001  001176             MOV     #1,$TMP1                ;RANGE OF PATTERN
1030 017172  004737  042174                     JSR     PC,GENBUF
1031 017176  000410                            BR      9$                      ;SKIP TO WRITE OPERATION
1032
1033                                    ;;*****************************************************************
1034                                    ;LOOP #2        WRITE,WRITE CHECK
1035
1036 017200                             8$:
1037
1038
1039                                    ;PREPARE DEVICE FOR WRITE OPERATION
1040 017200  004737  037316                     JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     017204  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
     017206  000404                            BR      9$              ;GO TO 9$ IF NO ERROR
     017210  000240                            NOP                     ;RETURN HERE IF ERROR
     017212  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     017214  000137  017634                     JMP     24$             ;GO TO 24$ IF ERROR
1041 017220                             9$:
1042
1043                                    ;WRITE DATA TO THE DRIVE
1044 017220                             10$:
1045 017220  012737  000061  001412             MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
```

```
1046 017226  012702  001553                    MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1047 017232  112722  000006                    MOVB    #RMDA,(R2)+
1048 017236  112722  000034                    MOVB    #RMDC,(R2)+
1049 017242  112722  000032                    MOVB    #RMOF,(R2)+
1050 017246  112722  000004                    MOVB    #RMBA,(R2)+
1051 017252  112722  000002                    MOVB    #RMWC,(R2)+
1052 017256  112722  000000                    MOVB    #RMCS1,(R2)+
1053 017262  112722  000200                    MOVB    #200,(R2)+              ;TERMINATE TABLE
1054
1055 017266  004737  043330                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     017272  000404                            BR      11$             ;GO TO 11$ IF NO ERROR
     017274  000240                            NOP                     ;RETURN HERE IF ERROR
     017276  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     017300  000137  017634                    JMP     24$             ;GO TO 24$ IF ERROR
1056 017304                            11$:
1057
1058                                   ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     017304  004737  042774                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1059
1060                                   ;WAIT FOR COMMAND TO COMPLETE
     017310  004737  043672                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1061
1062                                   ;GO READ STATUS FOR WRITE COMMAND
1063 017314  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     017320  000404                            BR      12$             ;GO TO 12$ IF NO ERROR
     017322  000240                            NOP                     ;RETURN HERE IF ERROR
     017324  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     017326  000137  017634                    JMP     24$             ;GO TO 24$ IF ERROR
1064 017332                            12$:
1065
1066                                   ;CHECK FOR ERRORS DURING WRITE OPERATION
1067 017332  004737  044056                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     017336  000405                            BR      13$             ;GO TO 13$ IF NO ERROR
     017340  000240                            NOP                     ;RETURN HERE IF ERROR
     017342  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     017344  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017346  000137  017634                    JMP     24$             ;GO TO 24$ IF ERROR
1068 017352                            13$:
1069 017352  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     017356  000405                            BR      14$             ;GO TO 14$ IF NO ERROR
     017360  000240                            NOP                     ;RETURN HERE IF ERROR
     017362  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     017364  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017366  000137  017634                    JMP     24$             ;GO TO 24$ IF ERROR
1070 017372                            14$:
1071 017372  004737  044710                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     017376  000405                            BR      15$             ;GO TO 15$ IF NO ERROR
     017400  000240                            NOP                     ;RETURN HERE IF ERROR
     017402  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     017404  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017406  000137  017634                    JMP     24$             ;GO TO 24$ IF ERROR
1072 017412                            15$:
1073
1074                                   ;CHANGE LOOP ADDRESSES
1075 017412  012737  017422  001124            MOV     #16$,$LPERR
1076 017420  000410                            BR      17$                     ;SKIP TO NEXT OPERATION
1077
```

```
                                        ;;*****************************************************************
                                        ;LOOP #3        WRITE CHECK

                                        16$:

                                        ;PREPARE DEVICE FOR READ OPERATION
017422  004737  037316                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
017426  154130                                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
017430  000404                                  BR      17$             ;GO TO 17$ IF NO ERROR
017432  000240                                  NOP                     ;RETURN HERE IF ERROR
017434  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
017436  000137  017634                          JMP     24$             ;GO TO 24$ IF ERROR
017442                                  17$:

                                        ;READ DATA FROM DEVICE
017442                                  18$:
017442  012737  000051  001412                  MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
017450  012702  001553                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
017454  112722  000006                          MOVB    #RMDA,(R2)+
017460  112722  000032                          MOVB    #RMOF,(R2)+
017464  112722  000034                          MOVB    #RMDC,(R2)+
017470  112722  000004                          MOVB    #RMBA,(R2)+
017474  112722  000002                          MOVB    #RMWC,(R2)+
017500  112722  000000                          MOVB    #RMCS1,(R2)+
017504  112712  000200                          MOVB    #200,(R2)

017510  004737  043330                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
017514  000404                                  BR      19$             ;GO TO 19$ IF NO ERROR
017516  000240                                  NOP                     ;RETURN HERE IF ERROR
017520  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
017522  000137  017634                          JMP     24$             ;GO TO 24$ IF ERROR
017526                                  19$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
017526  004737  042774                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
017532  004737  043672                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR READ OPERATION
017536  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
017542  000404                                  BR      20$             ;GO TO 20$ IF NO ERROR
017544  000240                                  NOP                     ;RETURN HERE IF ERROR
017546  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
017550  000137  017634                          JMP     24$             ;GO TO 24$ IF ERROR
017554                                  20$:

                                        ;CHECK FOR ERRORS DURING READ OPERATION
017554  004737  044056                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
017560  000405                                  BR      21$             ;GO TO 21$ IF NO ERROR
```

```
        017562  000240                          NOP                     ;RETURN HERE IF ERROR
        017564  104000                          EMT                     ;ERROR # DEFINED BY DSKERR SUBROUTINE
        017566  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017570  000137  017634                  JMP     24$             ;GO TO 24$ IF ERROR
1112    017574                          21$:
1113    017574  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        017600  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
        017602  000240                          NOP                     ;RETURN HERE IF ERROR
        017604  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        017606  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017610  000137  017634                  JMP     24$             ;GO TO 24$ IF ERROR
1114    017614                          22$:
1115    017614  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        017620  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
        017622  000240                          NOP                     ;RETURN HERE IF ERROR
        017624  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        017626  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017630  000137  017634                  JMP     24$             ;GO TO 24$ IF ERROR
1116    017634                          23$:
1117
1118    017634                          24$:
1119
1120                                    ;;*******************************************************************
                                        ;*TEST 11        WRITE, READ W/ IMPLIED SEEK
                                        ;;*******************************************************************
        017634                          TST11:
        017634  000004                          SCOPE                   ;SCOPE CALL
        017636  000240                          NOP                     ;START OF TEST
        017640  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        017644  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        017650  013701  001466                  MOV     TSTOUE,R1       ;(R1) = DEVICE BEING TESTED
        017654  012737  000011  001226          MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1121
1122                                    ;;*******************************************************************
1123                                    ;LOOP #1        FORMAT,WRITE,READ
1124
1125    017662                          1$:
1126
1127                                    ;PREPARE THE DEVICE FOR FORMAT OPERATION
1128    017662  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        017666  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        017670  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
        017672  000240                          NOP                     ;RETURN HERE IF ERROR
        017674  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        017676  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1129
1130                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1131    017702                          2$:
1132    017702  012737  000000  001446          MOV     #0,RMDCO                ;CYLINDER = 0
1133    017710  012737  000000  001420          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
```

```
1134 017716 013737 001446 021056          MOV     RMDC0,32$               ;SAVE DESIRED CYLINDER
1135 017724 012737 106436 001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
1136 017732 012737 177376 001414          MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
1137 017740 012737 010000 001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
1138 017746 012737 000063 001412          MOV     #WHIGO,RMCS10           ;WRITE HEADER AND DATA COMMAND
1139
1140                                ;VERIFY THAT SECTOR IS NOT BAD
     017754 004737 040242                 JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     017760 000405                        BR      3$              ;GO TO 3$ IF NO ERROR
     017762 104401 070426                 TYPE    ,STMSG          ;TYPE BAD SECTOR MESSAGE
     017766 104000                        EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     017770 000137 021054                 JMP     31$             ;GO TO 31$ IF ERROR
1141 017774                         3$:
1142 017774 012737 072016 001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
1143 020002 012737 000001 001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
1144 020010 004737 042174                 JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
1145
1146                                ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
1147 020014 012702 001553                 MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
1148 020020 112722 000034                 MOVB    #RMDC,(R2)+
1149 020024 112722 000006                 MOVB    #RMDA,(R2)+
1150 020030 112722 000004                 MOVB    #RMBA,(R2)+
1151 020034 112722 000002                 MOVB    #RMWC,(R2)+
1152 020040 112722 000032                 MOVB    #RMOF,(R2)+
1153 020044 112722 000000                 MOVB    #RMCS1,(R2)+
1154 020050 112712 000200                 MOVB    #200,(R2)               ;TERMINATE TABLE
1155 020054                         4$:
1156
1157                                ;FORMAT THE DRIVE
1158 020054 004737 043330                 JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020060 000404                        BR      5$              ;GO TO 5$ IF NO ERROR
     020062 000240                        NOP                     ;RETURN HERE IF ERROR
     020064 104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020066 000137 021054                 JMP     31$             ;GO TO 31$ IF ERROR
1159 020072                         5$:
1160
1161                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     020072 004737 042774                 JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1162
1163                                ;WAIT FOR COMMAND TO COMPLETE
     020076 004737 043672                 JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1164
1165                                ;GO READ STATUS FOR FORMAT OPERATION
1166 020102 004737 043060                 JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020106 000404                        BR      6$              ;GO TO 6$ IF NO ERROR
     020110 000240                        NOP                     ;RETURN HERE IF ERROR
     020112 104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020114 000137 021054                 JMP     31$             ;GO TO 31$ IF ERROR
1167 020120                         6$:
1168
1169                                ;VERIFY NO ERRORS DURING FORMAT
1170 020120 004737 056572                 JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     020124 000405                        BR      7$              ;GO TO 7$ IF NO ERROR
     020126 000240                        NOP                     ;RETURN HERE IF ERROR
     020130 104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     020132 004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020134 000137 021054                 JMP     31$             ;GO TO 31$ IF ERROR
```

```
1171 020140                          7$:
1172
1173                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1174 020140  012737  020150  001124          MOV     #8$,$LPERR
1175 020146  000410                          BR      9$                      ;SKIP TO WRITE OPERATION
1176
1177                                 ;;*****************************************************************
1178                                 ;LOOP #2        WRITE,READ
1179
1180 020150                          8$:
1181
1182                                 ;PREPARE DEVICE FOR WRITE OPERATION
1183 020150  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     020154  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                     ;VERIFY RECALIBRATION
     020156  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
     020160  000240                          NOP                     ;RETURN HERE IF ERROR
     020162  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     020164  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1184 020170                          9$:
1185 020170  012737  001466  001446          MOV     #822.,RMDCO             ;SEEK TO LAST CYLINDER
1186 020176  012737  000005  001412          MOV     #SEEK!GO,RMCS10         ;WRITE SEEK COMMAND
1187
1188 020204  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020210  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
     020212  000240                          NOP                     ;RETURN HERE IF ERROR
     020214  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020216  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1189 020222                          10$:
1190
1191                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     020222  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1192
1193                                 ;WAIT FOR COMMAND TO COMPLETE
     020226  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1194
1195                                 ;GO GET REGISTER STATUS
1196 020232  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020236  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
     020240  000240                          NOP                     ;RETURN HERE IF ERROR
     020242  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020244  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1197 020250                          11$:
1198
1199                                 ;VERIFY RESULTS OF SEEK COMMAND
1200 020250  004737  051174                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     020254  000405                          BR      12$             ;GO TO 12$ IF NO ERROR
     020256  000240                          NOP                     ;RETURN HERE IF ERROR
     020260  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     020262  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020264  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
```

```
1201 020270                          12$:
1202
1203                                 ;SETUP PARAMETERS AND EXCUTE WRITE DATA COMMAND
1204 020270                          13$:
1205 020270  013737  021056  001446        MOV     32$,RMDCO               ;RESTORE CYLINDER
1206 020276  012737  177400  001414        MOV     #-256.,RMWCO            ;CHANGE WORD COUNT
1207 020304  012737  106442  001416        MOV     #BUFONE+4,RMBAO         ;CHANGE BUS ADDRESS
1208 020312  012737  000061  001412        MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
1209 020320  012702  001553                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1210 020324  112722  000006                MOVB    #RMDA,(R2)+
1211 020330  112722  000034                MOVB    #RMDC,(R2)+
1212 020334  112722  000032                MOVB    #RMOF,(R2)+
1213 020340  112722  000004                MOVB    #RMBA,(R2)+
1214 020344  112722  000002                MOVB    #RMWC,(R2)+
1215 020350  112722  000000                MOVB    #RMCS1,(R2)+
1216 020354  112722  000200                MOVB    #200,(R2)+              ;TERMINATE TABLE
1217
1218 020360  004737  043330                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020364  000404                        BR      14$             ;GO TO 14$ IF NO ERROR
     020366  000240                        NOP                     ;RETURN HERE IF ERROR
     020370  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020372  000137  021054                JMP     31$             ;GO TO 31$ IF ERROR
1219 020376                          14$:
1220
1221                                 ;WAIT FOR COMMAND TO COMPLETE
     020376  004737  043672                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1222
1223                                 ;GO READ STATUS FOR WRITE COMMAND
1224 020402  004737  043060                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020406  000404                        BR      15$             ;GO TO 15$ IF NO ERROR
     020410  000240                        NOP                     ;RETURN HERE IF ERROR
     020412  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020414  000137  021054                JMP     31$             ;GO TO 31$ IF ERROR
1225 020420                          15$:
1226
1227                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
1228 020420  004737  044056                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     020424  000405                        BR      16$             ;GO TO 16$ IF NO ERROR
     020426  000240                        NOP                     ;RETURN HERE IF ERROR
     020430  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     020432  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020434  000137  021054                JMP     31$             ;GO TO 31$ IF ERROR
1229 020440                          16$:
1230 020440  004737  056572                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     020444  000405                        BR      17$             ;GO TO 17$ IF NO ERROR
     020446  000240                        NOP                     ;RETURN HERE IF ERROR
     020450  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     020452  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020454  000137  021054                JMP     31$             ;GO TO 31$ IF ERROR
1231 020460                          17$:
1232 020460  004737  044710                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     020464  000405                        BR      18$             ;GO TO 18$ IF NO ERROR
     020466  000240                        NOP                     ;RETURN HERE IF ERROR
     020470  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     020472  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020474  000137  021054                JMP     31$             ;GO TO 31$ IF ERROR
1233 020500                          18$:
```

```
1234
1235                                    ;CHANGE LOOP ADDRESSES
1236 020500  012737  020510  001124             MOV     #19$,$LPERR
1237 020506  000410                             BR      20$                     ;SKIP TO NEXT OPERATION
1238
1239                                    ;;*****************************************************************
1240                                    ;LOOP #3        READ
1241
1242 020510                             19$:
1243
1244                                    ;PREPARE DEVICE FOR READ OPERATION
1245 020510  004737  037316                     JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     020514  154130                             .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
     020516  000404                             BR      20$             ;GO TO 20$ IF NO ERROR
     020520  000240                             NOP                     ;RETURN HERE IF ERROR
     020522  104000                             EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     020524  000137  021054                     JMP     31$             ;GO TO 31$ IF ERROR
1246 020530                             20$:
1247 020530  012737  001466  001446             MOV     #822.,RMDCO             ;SEEK TO LAST CYLINDER
1248 020536  012737  000005  001412             MOV     #SEEK!GO,RMCS1O         ;WRITE SEEK COMMAND
1249
1250 020544  004737  043330                     JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020550  000404                             BR      21$             ;GO TO 21$ IF NO ERROR
     020552  000240                             NOP                     ;RETURN HERE IF ERROR
     020554  104000                             EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020556  000137  021054                     JMP     31$             ;GO TO 31$ IF ERROR
1251 020562                             21$:
1252
1253                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     020562  004737  042774                     JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1254
1255                                    ;WAIT FOR COMMAND TO COMPLETE
     020566  004737  043672                     JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1256
1257                                    ;GO GET REGISTER STATUS
1258 020572  004737  043060                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020576  000404                             BR      22$             ;GO TO 22$ IF NO ERROR
     020600  000240                             NOP                     ;RETURN HERE IF ERROR
     020602  104000                             EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020604  000137  021054                     JMP     31$             ;GO TO 31$ IF ERROR
1259 020610                             22$:
1260
1261                                    ;VERIFY RESUTS OF SEEK COMMAND
1262 020610  004737  051174                     JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     020614  000405                             BR      23$             ;GO TO 23$ IF NO ERROR
     020616  000240                             NOP                     ;RETURN HERE IF ERROR
     020620  104000                             EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     020622  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020624  000137  021054                     JMP     31$             ;GO TO 31$ IF ERROR
1263 020630                             23$:
```

```
1264
1265                                    ;SETUP PARAMETERS AND EXCUTE READ DATA COMMAND
1266 020630                             24$:
1267 020630  013737  021056  001446          MOV     32$,RMDCO               ;RESTORE CYLINDER
1268 020636  012737  000071  001412          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
1269 020644  012737  107446  001416          MOV     #BUFTWO+4,RMBAO         ;LOAD STARTING BUFFER ADDRESS
1270 020652  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1271 020656  112722  000006                  MOVB    #RMDA,(R2)+
1272 020662  112722  000032                  MOVB    #RMOF,(R2)+
1273 020666  112722  000034                  MOVB    #RMDC,(R2)+
1274 020672  112722  000004                  MOVB    #RMBA,(R2)+
1275 020676  112722  000002                  MOVB    #RMWC,(R2)+
1276 020702  112722  000000                  MOVB    #RMCS1,(R2)+
1277 020706  112712  000200                  MOVB    #200,(R2)
1278
1279 020712  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020716  000404                          BR      25$             ;GO TO 25$ IF NO ERROR
     020720  000240                          NOP                     ;RETURN HERE IF ERROR
     020722  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020724  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1280 020730                             25$:
1281
1282                                    ;WAIT FOR COMMAND TO COMPLETE
     020730  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1283
1284                                    ;GO READ STATUS FOR READ OPERATION
1285 020734  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020740  000404                          BR      26$             ;GO TO 26$ IF NO ERROR
     020742  000240                          NOP                     ;RETURN HERE IF ERROR
     020744  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020746  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1286 020752                             26$:
1287
1288                                    ;CHECK FOR ERRORS DURING READ OPERATION
1289 020752  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     020756  000405                          BR      27$             ;GO TO 27$ IF NO ERROR
     020760  000240                          NOP                     ;RETURN HERE IF ERROR
     020762  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     020764  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020766  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1290 020772                             27$:
1291 020772  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     020776  000405                          BR      28$             ;GO TO 28$ IF NO ERROR
     021000  000240                          NOP                     ;RETURN HERE IF ERROR
     021002  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     021004  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021006  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1292 021012                             28$:
1293 021012  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     021016  000405                          BR      29$             ;GO TO 29$ IF NO ERROR
     021020  000240                          NOP                     ;RETURN HERE IF ERROR
     021022  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     021024  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021026  000137  021054                  JMP     31$             ;GO TO 31$ IF ERROR
1294 021032                             29$:
1295 021032  004737  042432                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
     021036  106442                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
```

```
        021040  107446                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        021042  000404                          BR      30$                     ;GO TO 30$ IF NO ERROR
        021044  000240                          NOP                             ;RETURN HERE IF ERROR
        021046  104000                          EMT                             ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        021050  000137  021054                  JMP     31$                     ;GO TO 31$ IF ERROR
  1296  021054                          30$:
  1297
  1298  021054                          31$:
  1299  021054  000401                          BR      33$
  1300
  1301  021056  000000                  32$:    .WORD   0
  1302
  1303  021060                          33$:
  1304
  1305                                  ;;*******************************************************************
                                        ;*TEST 12        WRITE, WRITE CHECK W/ HEAD SWITCHING
                                        ;;*******************************************************************
        021060                          TST12:
        021060  000004                          SCOPE                           ;SCOPE CALL
        021062  000240                          NOP                             ;START OF TEST
        021064  012706  001100                  MOV     #STACK,SP               ;INITIALIZE STACK POINTER
        021070  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
        021074  013701  001466                  MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
        021100  012737  000012  001226          MOV     #12,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
  1306
  1307                                  ;;*******************************************************************
  1308                                  ;LOOP #1        FORMAT,WRITE,WRITE CHECK
  1309
  1310                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  1311  021106                          1$:
  1312  021106  012737  000000  001446          MOV     #0,RMDC0                ;CYLINDER = 0
  1313  021114  112737  000000  001421          MOVB    #0,RMDA0+1              ;TRACK   0
  1314  021122  112737  000037  001420  2$:     MOVB    #31.,RMDA0              ;SECTOR = 31.
  1315  021130  012737  010000  001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
  1316  021136  012737  106436  001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
  1317  021144  012737  176774  001414          MOV     #-258.*2,RMWC0          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
  1318  021152  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
  1319
  1320                                  ;VERIFY THAT SECTOR IS NOT BAD
        021160  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        021164  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        021166  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        021172  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        021174  000137  022266                  JMP     30$             ;GO TO 30$ IF ERROR
  1321  021200                          3$:
  1322  021200  123727  001420  000037          CMPB    RMDA0,#31.              ;IS LAST TRACK ASSIGNED ?
  1323  021206  001345                          BNE     2$                      ;BR IF NO
  1324  021210  012737  072016  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
  1325  021216  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
  1326  021224  004737  042174                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
  1327
  1328                                  ;PREPARE THE DEVICE FOR FORMAT OPERATION
  1329  021230  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        021234  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
```

```
                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                        ;VERIFY PACK ACKNOWLEDGE
                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                        ;VERIFY RECALIBRATION
 021236  000404                       BR      4$        ;GO TO 4$ IF NO ERROR
 021240  000240                       NOP               ;RETURN HERE IF ERROR
 021242  104000                       EMT               ;ERROR # DEFINED BY TSTPRP SUBROUTINE
 021244  000137  022266               JMP     30$       ;GO TO 30$ IF ERROR
1330 021250                    4$:
1331
1332                           ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
1333 021250  012702  001553           MOV     #PUTINX,R2          ;R2 - BYTE ENTRY POSITION
1334 021254  112722  000034           MOVB    #RMDC,(R2)+
1335 021260  112722  000006           MOVB    #RMDA,(R2)+
1336 021264  112722  000004           MOVB    #RMBA,(R2)+
1337 021270  112722  000002           MOVB    #RMWC,(R2)+
1338 021274  112722  000032           MOVB    #RMOF,(R2)+
1339 021300  112722  000000           MOVB    #RMCS1,(R2)+
1340 021304  112712  000200           MOVB    #200,(R2)           ;TERMINATE TABLE
1341
1342                           ;FORMAT THE DRIVE
1343 021310  004737  043330           JSR     PC,PUT    ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
 021314  000404                       BR      5$        ;GO TO 5$ IF NO ERROR
 021316  000240                       NOP               ;RETURN HERE IF ERROR
 021320  104000                       EMT               ;ERROR # DEFINED BY PUT SUBROUTINE
 021322  000137  022266               JMP     30$       ;GO TO 30$ IF ERROR
1344 021326                    5$:
1345
1346                           ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
 021326  004737  042774           JSR     PC,GETSTS     ;GO TO GETSTS SUBROUTINE
1347
1348                           ;WAIT FOR COMMAND TO COMPLETE
 021332  004737  043672           JSR     PC,TIMOUT     ;GO TO TIMOUT SUBROUTINE
1349
1350                           ;GO READ STATUS FOR FORMAT OPERATION
1351 021336  004737  043060           JSR     PC,GET    ;GO READ REGISTER(S) WITH GET SUBROUTINE
 021342  000404                       BR      6$        ;GO TO 6$ IF NO ERROR
 021344  000240                       NOP               ;RETURN HERE IF ERROR
 021346  104000                       EMT               ;ERROR # DEFINED BY GET SUBROUTINE
 021350  000137  022266               JMP     30$       ;GO TO 30$ IF FRROR
1352 021354                    6$:
1353
1354                           ;VERIFY NO ERRORS DURING FORMAT
1355 021354  004737  056572           JSR     PC,DTASTS     ;GO VERIFY RESULTS OF DATA TRANSFER
 021360  000405                       BR      7$        ;GO TO 7$ IF NO ERROR
 021362  000240                       NOP               ;RETURN HERE IF ERROR
 021364  104000                       EMT               ;ERROR # DEFINED BY DTASTS SUBROUTINE
 021366  004736                       JSR     PC,@(SP)+     ;GO BACK FOR MORE ERROR CHECKS
 021370  000137  022266               JMP     30$       ;GO TO 30$ IF ERROR
1356 021374                    7$:
1357
1358                           ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1359 021374  012737  021446  001124       MOV     #8$,$LPERR
1360
1361                           ;REGENERATE BUFFER
1362 021402  012737  177000  001414       MOV     #-256.*2,RMWCO      ;CHANGE WORD COUNT
1363 021410  012737  106436  001416       MOV     #BUFONE,RMBAO       ;CHANGE BUS ADDRESS
```

```
1364 021416  012737  072060  001174          MOV     #ZEROS,$TMP0            ;STARTING ADDRESS
1365 021424  012737  000001  001176          MOV     #1,$TMP1                ;RANGE
1366 021432  012737  000060  001412          MOV     #WD,RMCS1O              ;WRITE DATA
1367 021440  004737  042174                  JSR     PC,GENBUF               ;GENERATE BUFFER
1368 021444  000410                          BR      9$                      ;SKIP TO WRITE OPERATION
1369
1370                                 ;;*****************************************************************
1371                                 ;LOOP #2        WRITE,WRITE CHECK
1372
1373 021446                          8$:
1374
1375                                 ;PREPARE DEVICE FOR WRITE OPERATION
1376 021446  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     021452  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                     ;VERIFY RECALIBRATION
     021454  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
     021456  000240                          NOP                     ;RETURN HERE IF ERROR
     021460  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     021462  000137  022266                  JMP     30$             ;GO TO 30$ IF ERROR
1377 021466                          9$:
1378 021466  012737  000005  001412          MOV     #SEEK!GO,RMCS1O         ;LOAD SEEK COMMAND
1379
1380 021474  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021500  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
     021502  000240                          NOP                     ;RETURN HERE IF ERROR
     021504  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021506  000137  022266                  JMP     30$             ;GO TO 30$ IF ERROR
1381 021512                          10$:
1382
1383                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     021512  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1384
1385                                 ;WAIT FOR COMMAND TO COMPLETE
     021516  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1386
1387                                 ;GO GET REGISTER STATUS
1388 021522  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021526  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
     021530  000240                          NOP                     ;RETURN HERE IF ERROR
     021532  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021534  000137  022266                  JMP     30$             ;GO TO 30$ IF ERROR
1389 021540                          11$:
1390
1391                                 ;VERIFY RESULTS OF SEEK COMMAND
1392 021540  004737  051174                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     021544  000405                          BR      12$             ;GO TO 12$ IF NO ERROR
     021546  000240                          NOP                     ;RETURN HERE IF ERROR
     021550  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     021552  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021554  000137  022266                  JMP     30$             ;GO TO 30$ IF ERROR
1393 021560                          12$:
```

```
1394
1395                                        ;WRITE DATA TO THE DRIVE
1396 021560                                13$:
1397 021560  012737  000061  001412              MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
1398 021566  012702  001553                      MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1399 021572  112722  000006                      MOVB    #RMDA,(R2)+
1400 021576  112722  000034                      MOVB    #RMDC,(R2)+
1401 021602  112722  000032                      MOVB    #RMOF,(R2)+
1402 021606  112722  000004                      MOVB    #RMBA,(R2)+
1403 021612  112722  000002                      MOVB    #RMWC,(R2)+
1404 021616  112722  000000                      MOVB    #RMCS1,(R2)+
1405 021622  112722  000200                      MOVB    #200,(R2)+              ;TERMINATE TABLE
1406
1407 021626  004737  043330                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021632  000404                              BR      14$             ;GO TO 14$ IF NO ERROR
     021634  000240                              NOP                     ;RETURN HERE IF ERROR
     021636  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021640  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1408 021644                                14$:
1409
1410                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     021644  004737  042774                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1411
1412                                        ;WAIT FOR COMMAND TO COMPLETE
     021650  004737  043672                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1413
1414                                        ;GO READ STATUS FOR WRITE COMMAND
1415 021654  004737  043060                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021660  000404                              BR      15$             ;GO TO 15$ IF NO ERROR
     021662  000240                              NOP                     ;RETURN HERE IF ERROR
     021664  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021666  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1416 021672                                15$:
1417
1418                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
1419 021672  004737  044056                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     021676  000405                              BR      16$             ;GO TO 16$ IF NO ERROR
     021700  000240                              NOP                     ;RETURN HERE IF ERROR
     021702  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     021704  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021706  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1420 021712                                16$:
1421 021712  004737  056572                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     021716  000405                              BR      17$             ;GO TO 17$ IF NO ERROR
     021720  000240                              NOP                     ;RETURN HERE IF ERROR
     021722  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     021724  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021726  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1422 021732                                17$:
1423 021732  004737  044710                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     021736  000405                              BR      18$             ;GO TO 18$ IF NO ERROR
     021740  000240                              NOP                     ;RETURN HERE IF ERROR
     021742  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     021744  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021746  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1424 021752                                18$:
1425
```

```
                                        ;CHANGE LOOP ADDRESSES
1427 021752  012737  021762  001124              MOV     #19$,$LPERR
1428 021760  000410                              BR      20$                     ;SKIP TO NEXT OPERATION
1429
1430                                    ;;*****************************************************************
1431                                    ;LOOP #3        WRITE CHECK
1432
1433 021762                             19$:
1434
1435                                    ;PREPARE DEVICE FOR READ OPERATION
1436 021762  004737  037316                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     021766  154130                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                         ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                         ;CLEAR CONTROLLER & SELECT DEVICE
                                                                         ;VERIFY CONTROLLER CLEAR OPERATION
                                                                         ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                         ;VERIFY PACK ACKNOWLEDGE
                                                                         ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                         ;VERIFY RECALIBRATION
     021770  000404                              BR      20$             ;GO TO 20$ IF NO ERROR
     021772  000240                              NOP                     ;RETURN HERE IF ERROR
     021774  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     021776  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1437 022002                             20$:
1438 022002  012737  000005  001412              MOV     #SEEK!GO,RMCS10         ;LOAD SEEK COMMAND
1439
1440 022010  004737  043330                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022014  000404                              BR      21$             ;GO TO 21$ IF NO ERROR
     022016  000240                              NOP                     ;RETURN HERE IF ERROR
     022020  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     022022  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1441 022026                             21$:
1442
1443                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     022026  004737  042774                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1444
1445                                    ;WAIT FOR COMMAND TO COMPLETE
     022032  004737  043672                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1446
1447                                    ;GO READ REGISTER STATUS
1448 022036  004737  043060                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022042  000404                              BR      22$             ;GO TO 22$ IF NO ERROR
     022044  000240                              NOP                     ;RETURN HERE IF ERROR
     022046  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     022050  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1449 022054                             22$:
1450
1451                                    ;GO VERIFY RESULTS OF SEEK
1452 022054  004737  051174                      JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     022060  000405                              BR      23$             ;GO TO 23$ IF NO ERROR
     022062  000240                              NOP                     ;RETURN HERE IF ERROR
     022064  104000                              EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     022066  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022070  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1453 022074                             23$:
1454
1455                                    ;READ DATA FROM DEVICE
```

```
1456 022074                                24$:
1457 022074  012737  000051  001412              MOV     #WCD.GO,RMCS10          ;READ DATA COMMAND
1458 022102  012702  001553                      MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1459 022106  112722  000006                      MOVB    #RMDA,(R2)+
1460 022112  112722  000032                      MOVB    #RMOF,(R2)+
1461 022116  112722  000034                      MOVB    #RMDC,(R2)+
1462 022122  112722  000004                      MOVB    #RMBA,(R2)+
1463 022126  112722  000002                      MOVB    #RMWC,(R2)+
1464 022132  112722  000000                      MOVB    #RMCS1,(R2)+
1465 022136  112712  000200                      MOVB    #200,(R2)
1466
1467 022142  004737  043330                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022146  000404                              BR      25$             ;GO TO 25$ IF NO ERROR
     022150  000240                              NOP                     ;RETURN HERE IF ERROR
     022152  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     022154  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1468 022160                                25$:
1469
1470                                       ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     022160  004737  042774                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1471
1472                                       ;WAIT FOR COMMAND TO COMPLETE
     022164  004737  043672                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1473
1474                                       ;GO READ STATUS FOR READ OPERATION
1475 022170  004737  043060                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022174  000404                              BR      26$             ;GO TO 26$ IF NO ERROR
     022176  000240                              NOP                     ;RETURN HERE IF ERROR
     022200  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     022202  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1476 022206                                26$:
1477
1478                                       ;CHECK FOR ERRORS DURING READ OPERATION
1479 022206  004737  044056                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     022212  000405                              BR      27$             ;GO TO 27$ IF NO ERROR
     022214  000240                              NOP                     ;RETURN HERE IF ERROR
     022216  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     022220  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022222  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1480 022226                                27$:
1481 022226  004737  056572                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     022232  000405                              BR      28$             ;GO TO 28$ IF NO ERROR
     022234  000240                              NOP                     ;RETURN HERE IF ERROR
     022236  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     022240  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022242  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1482 022246                                28$:
1483 022246  004737  044710                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     022252  000405                              BR      29$             ;GO TO 29$ IF NO ERROR
     022254  000240                              NOP                     ;RETURN HERE IF ERROR
     022256  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     022260  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022262  000137  022266                      JMP     30$             ;GO TO 30$ IF ERROR
1484 022266                                29$:
1485
1486 022266                                30$:
1487
```

```
1488                                    ;;*****************************************************************
                                        ;*TEST 13       WRITE, WRITE CHECK W/ MID-TRANSFER SEEK
                                        ;;*****************************************************************
        022266                          TST13:
        022266  000004                          SCOPE                   ;SCOPE CALL
        022270  000240                          NOP                     ;START OF TEST
        022272  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        022276  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        022302  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        022306  012737  000013  001226          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1489
1490                                    ;;*****************************************************************
1491                                    ;LOOP #1        FORMAT,WRITE,WRITE CHECK
1492
1493                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1494 022314                            1$:
1495 022314  012737  000000  001446          MOV     #0,RMDC0                ;CYLINDER = 0
1496 022322  013737  001334  001420  2$:     MOV     LSTRK,RMDA0             ;SET LAST TRACK AND
1497 022330  112737  000037  001420          MOVB    #31.,RMDA0              ;LAST SECTOR
1498 022336  012737  010000  001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
1499 022344  012737  106436  001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
1500 022352  012737  176774  001414          MOV     #-258.*2,RMWC0          ;WORD COUNT FOR 2 SECTORS (2' COMP)
1501 022360  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
1502
1503                                    ;VERIFY THAT SECTOR IS NOT BAD
        022366  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        022372  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        022374  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        022400  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        022402  000137  023310                  JMP     25$             ;GO TO 25$ IF ERROR
1504 022406                             3$:
1505 022406  123737  001421  001335          CMPB    RMDA0+1,LSTRK+1         ;IS LAST TRACK ASSIGNED ?
1506 022414  001342                          BNE     2$                      ;BR IF NO
1507 022416  012737  072060  001174          MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
1508 022424  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
1509 022432  004737  042174                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
1510
1511                                    ;PREPARE THE DEVICE FOR FORMAT OPERATION
1512 022436  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        022442  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        022444  000404                          BR      4$              ;GO TO 4$ IF NO ERROR
        022446  000240                          NOP                     ;RETURN HERE IF ERROR
        022450  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        022452  000137  023310                  JMP     25$             ;GO TO 25$ IF ERROR
1513 022456                             4$:
1514
1515                                    ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
1516 022456  012702  001553                  MOV     #PUTINX,R2              ;R2 - BYTE ENTRY POSITION
1517 022462  112722  000034                  MOVB    #RMDC,(R2)+
1518 022466  112722  000006                  MOVB    #RMDA,(R2)+
```

```
1519 022472  112722  000004              MOVB    #RMBA,(R2)+
1520 022476  112722  000002              MOVB    #RMWC,(R2)+
1521 022502  112722  000032              MOVB    #RMOF,(R2)+
1522 022506  112722  000000              MOVB    #RMCS1,(R2)+
1523 022512  112712  000200              MOVB    #200,(R2)               ;TERMINATE TABLE
1524 022516                          5$:
1525 022516  004737  043330              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022522  000404                      BR      6$              ;GO TO 6$ IF NO ERROR
     022524  000240                      NOP                     ;RETURN HERE IF ERROR
     022526  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     022530  000137  023310              JMP     25$             ;GO TO 25$ IF ERROR
1526 022534                          6$:
1527
1528                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     022534  004737  042774              JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1529
1530                                 ;WAIT FOR COMMAND TO COMPLETE
     022540  004737  043672              JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1531
1532
1533 022544  004737  043060              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022550  000404                      BR      7$              ;GO TO 7$ IF NO ERROR
     022552  000240                      NOP                     ;RETURN HERE IF ERROR
     022554  104000                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     022556  00013   023310              JMP     25$             ;GO TO 25$ IF ERROR
1534 022562                          7$:
1535
1536                                 ;VERIFY NO ERRORS DURING FORMAT
1537 022562  004737  056572              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     022566  000405                      BR      8$              ;GO TO 8$ IF NO ERROR
     022570  000240                      NOP                     ;RETURN HERE IF ERROR
     022572  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     022574  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022576  000137  023310              JMP     25$             ;GO TO 25$ IF ERROR
1538 022602                          8$:
1539
1540                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1541 022602  012737  022654  001124        MOV     #9$,$LPERR
1542
1543                                 ;REGENERATE BUFFER
1544 022610  012737  000060  001412        MOV     #WD,RMCS10              ;WRITE DATA
1545 022616  012737  177000  001414        MOV     #-256.*2,RMWC0  ;CHANGE WORD COUNT
1546 022624  012737  106436  001416        MOV     #BUFONE,RMBA0           ;CHANGE BUS ADDRSS
1547 022632  012737  072016  001174        MOV     #ONES,$TMP0             ;PATTERN ADDRESS
1548 022640  012737  000001  001176        MOV     #1,$TMP1                ;PATTERN RANGE
1549 022646  004737  042174              JSR     PC,GENBUF
1550 022652  000410                      BR      10$                     ;SKIP TO WRITE OPERATION
1551
1552                                 ;;*****************************************************************
1553                                 ;LOOP #2        WRITE,WRITE CHECK
1554
1555 022654                          9$:
1556
1557                                 ;PREPARE DEVICE FOR WRITE OPERATION
1558 022654  004737  037316              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     022660  154130                      .WOR    154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                 ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
```

```
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                ;VERIFY RECALIBRATION
     022662  000404                    BR      10$              ;GO TO 10$ IF NO ERROR
     022664  000240                    NOP                      ;RETURN HERE IF ERROR
     022666  104000                    EMT                      ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     022670  000137  023310            JMP     25$              ;GO TO 25$ IF ERROR
1559 022674                        10$:
1560
1561                               ;WRITE DATA TO THE DRIVE
1562 022674                        1$:
1563 022674  012737  000061  001412  MOV     #WD!GO,RMCS1O            ;WRITE DATA COMMAND
1564 022702  012702  001553          MOV     #PUTINX,R2               ;LOAD PUT REGISTER INDEX TABLE
1565 022706  112722  000006          MOVB    #RMDA,(R2)+
1566 022712  112722  000034          MOVB    #RMDC,(R2)+
1567 022716  112722  000032          MOVB    #RMOF,(R2)+
1568 022722  112722  000004          MOVB    #RMBA,(R2)+
1569 022726  112722  000002          MOVB    #RMWC,(R2)+
1570 022732  112722  000000          MOVB    #RMCS1,(R2)+
1571 022736  112722  000200          MOVB    #200,(R2)+               ;TERMINATE TABLE
1572
1573 022742  004737  043330          JSR     PC,PUT           ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022746  000404                  BR      12$              ;GO TO 12$ IF NO ERROR
     022750  000240                  NOP                      ;RETURN HERE IF ERROR
     022752  104000                  EMT                      ;ERROR # DEFINED BY PUT SUBROUTINE
     022754  000137  023310          JMP     25$              ;GO TO 25$ IF ERROR
1574 022760                        12$:
1575
1576                               ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     022760  004737  042774          JSR     PC,GETSTS        ;GO TO GETSTS SUBROUTINE
1577
1578                               ;WAIT FOR COMMAND TO COMPLETE
     022764  004737  043672          JSR     PC,TIMOUT        ;GO TO TIMOUT SUBROUTINE
1579
1580                               ;GO READ STATUS FOR WRITE COMMAND
1581 022770  004737  043060          JSR     PC,GET           ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022774  000404                  BR      13$              ;GO TO 13$ IF NO ERROR
     022776  000240                  NOP                      ;RETURN HERE IF ERROR
     023000  104000                  EMT                      ;ERROR # DEFINED BY GET SUBROUTINE
     023002  000137  023310          JMP     25$              ;GO TO 25$ IF ERROR
1582 023006                        13$:
1583
1584                               ;CHECK FOR ERRORS DURING WRITE OPERATION
1585 023006  004737  044056          JSR     PC,PRIERR        ;GO CHECK FOR PRIMARY ERRORS
     023012  000405                  BR      14$              ;GO TO 14$ IF NO ERROR
     023014  000240                  NOP                      ;RETURN HERE IF ERROR
     023016  104000                  EMT                      ;ERROR # DEFINED BY PRIERR SUBROUTINE
     023020  004736                  JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
     023022  000137  023310          JMP     25$              ;GO TO 25$ IF ERROR
1586 023026                        14$:
1587 023026  004737  056572          JSR     PC,DTASTS        ;GO VERIFY RESULTS OF DATA TRANSFER
     023032  000405                  BR      15$              ;GO TO 15$ IF NO ERROR
     023034  000240                  NOP                      ;RETURN HERE IF ERROR
     023036  104000                  EMT                      ;ERROR # DEFINED BY DTASTS SUBROUTINE
```

```
        023040  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023042  000137  023310                  JMP     25$             ;GO TO 25$ IF ERROR
  1588  023046                          15$:
  1589  023046  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        023052  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        023054  000240                          NOP                     ;RETURN HERE IF ERROR
        023056  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        023060  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023062  000137  023310                  JMP     25$             ;GO TO 25$ IF ERROR
  1590  023066                          16$:
  1591
  1592                                  ;CHANGE LOOP ADDRESSES
  1593  023066  012737  023076  001124          MOV     #17$,$LPERR
  1594  023074  000410                          BR      18$                     ;SKIP TO NEXT OPERATION
  1595
  1596                                  ;;*****************************************************************
  1597                                  ;LOOP #3        WRITE CHECK
  1598
  1599  023076                          17$:
  1600
  1601                                  ;PREPARE DEVICE FOR READ OPERATION
  1602  023076  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        023102  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER-CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        023104  000404                          BR      18$             ;GO TO 18$ IF NO ERROR
        023106  000240                          NOP                     ;RETURN HERE IF ERROR
        023110  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        023112  000137  023310                  JMP     25$             ;GO TO 25$ IF ERROR
  1603  023116                          18$:
  1604
  1605                                  ;READ DATA FROM DEVICE
  1606  023116                          19$:
  1607  023116  012737  000051  001412          MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
  1608  023124  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  1609  023130  112722  000006                  MOVB    #RMDA,(R2)+
  1610  023134  112722  000032                  MOVB    #RMOF,(R2)+
  1611  023140  112722  000034                  MOVB    #RMDC,(R2)+
  1612  023144  112722  000004                  MOVB    #RMBA,(R2)+
  1613  023150  112722  000002                  MOVB    #RMWC,(R2)+
  1614  023154  112722  000000                  MOVB    #RMCS1,(R2)+
  1615  023160  112712  000200                  MOVB    #200,(R2)
  1616
  1617  023164  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        023170  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        023172  000240                          NOP                     ;RETURN HERE IF ERROR
        023174  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        023176  000137  023310                  JMP     25$             ;GO TO 25$ IF ERROR
  1618  023202                          20$:
  1619
  1620                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        023202  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
```

```
1621
1622                                    ;WAIT FOR COMMAND TO COMPLETE
023206  004737  043672                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1623
1624                                    ;GO READ STATUS FOR READ OPERATION
1625 023212  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
023216  000404                                 BR      21$             ;GO TO 21$ IF NO ERROR
023220  000240                                 NOP                     ;RETURN HERE IF ERROR
023222  104000                                 EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
023224  000137  023310                        JMP     25$             ;GO TO 25$ IF ERROR
1626 023230                             21$:
1627
1628                                    ;CHECK FOR ERRORS DURING READ OPERATION
1629 023230  004737  044056                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
023234  000405                                 BR      22$             ;GO TO 22$ IF NO ERROR
023236  000240                                 NOP                     ;RETURN HERE IF ERROR
023240  104000                                 EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
023242  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
023244  000137  023310                        JMP     25$             ;GO TO 25$ IF ERROR
1630 023250                             22$:
1631 023250  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
023254  000405                                 BR      23$             ;GO TO 23$ IF NO ERROR
023256  000240                                 NOP                     ;RETURN HERE IF ERROR
023260  104000                                 EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
023262  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
023264  000137  023310                        JMP     25$             ;GO TO 25$ IF ERROR
1632 023270                             23$:
1633 023270  004737  044710                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
023274  000405                                 BR      24$             ;GO TO 24$ IF NO ERROR
023276  000240                                 NOP                     ;RETURN HERE IF ERROR
023300  104000                                 EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
023302  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
023304  000137  023310                        JMP     25$             ;GO TO 25$ IF ERROR
1634 023310                             24$:
1635
1636 023310                             25$:
1637
1638                                    ;;*******************************************************************
                                        ;*TEST 14        WRITE, READ W/ HCE ERROR
                                        ;;*******************************************************************
023310                                  TST14:
023310  000004                                 SCOPE                   ;SCOPE CALL
023312  000240                                 NOP                     ;START OF TEST
023314  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
023320  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
023324  013701  001466                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
023330  012737  000014  001226               MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1639
1640                                    ;;*******************************************************************
1641                                    ;LOOP #1        FORMAT,WRITE,READ
1642
1643 023336  012704  000000                    MOV     #0,R4           ;R4 = 1ST HEADER WORD
1644 023342  012703  000001             1$:    MOV     #1,R3           ;R3 = HCE BIT
1645 023346                             2$:
1646
1647                                    ;PREPARE THE DEVICE FOR FORMAT OPERATION
1648 023346  004737  037316                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
```

*14 WRITE, READ W/ HCE ERROR

```
        023352  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        023354  000404                          BR      3$              ;GO TO 3$ IF NO ERROR
        023356  000240                          NOP                     ;RETURN HERE IF ERROR
        023360  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        023362  000137  025156                  JMP     46$             ;GO TO 46$ IF ERROR
  1649
  1650                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  1651 023366                           3$:
  1652 023366   012737  000000  001446          MOV     #0,RMDC0        ;CYLINDER = 0
  1653 023374   012737  000000  001420          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
  1654 023402   012737  106436  001416          MOV     #BUFONE,RMBA0   ;BUS ADDRESS
  1655 023410   012737  177376  001414          MOV     #-258.,RMWC0    ;2 + 256. WORDS (2'S COMP)
  1656 023416   012737  010000  001444          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
  1657 023424   012737  000063  001412          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA COMMAND
  1658
  1659                                  ;VERIFY THAT SECTOR IS NOT BAD
        023432  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        023436  000405                          BR      4$              ;GO TO 4$ IF NO ERROR
        023440  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        023444  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        023446  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
  1660 023452                           4$:
  1661 023452   012737  072060  001174          MOV     #ZEROS,$TMP0    ;STARTING ADDRESS OF PATTERN
  1662 023460   012737  000001  001176          MOV     #1,$TMP1        ;RANGE OF PATTERN
  1663 023466   004737  042174                  JSR     PC,GENBUF       ;GO GENERATE BUFFER FOR FORMAT
  1664
  1665                                  ;CHANGE HEADER WORD TO FORCE HCE DURING WRITE & READ
  1666 023472   030364  106436                  BIT     R3,BUFONE(R4)   ;SET OR RESET FOR HCE??
  1667 023476   001403                          BEQ     5$
  1668 023500   040364  106436                  BIC     R3,BUFONE(R4)   ;RESET BIT FOR HCE
  1669 023504   000402                          BR      6$
  1670 023506   050364  106436          5$:     BIS     R3,BUFONE(R4)   ;SET FOR HCE
  1671
  1672                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
  1673 023512                           6$:
  1674 023512   012702  001553                  MOV     #PUTINX,R2      ;R2 - BYTE ENTRY POSITION
  1675 023516   112722  000034                  MOVB    #RMDC,(R2)+
  1676 023522   112722  000006                  MOVB    #RMDA,(R2)+
  1677 023526   112722  000004                  MOVB    #RMBA,(R2)+
  1678 023532   112722  000002                  MOVB    #RMWC,(R2)+
  1679 023536   112722  000032                  MOVB    #RMOF,(R2)+
  1680 023542   112722  000000                  MOVB    #RMCS1,(R2)+
  1681 023546   112712  000200                  MOVB    #200,(R2)       ;TERMINATE TABLE
  1682 023552                           7$:
  1683 023552   004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        023556  000404                          BR      8$              ;GO TO 8$ IF NO ERROR
        023560  000240                          NOP                     ;RETURN HERE IF ERROR
        023562  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        023564  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
  1684 023570                           8$:
```

```
1685
1686                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
023570  004737  042774                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1687
1688                                    ;WAIT FOR COMMAND TO COMPLETE
023574  004737  043672                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1689
1690                                    ;GO READ STATUS FOR FORMAT OPERATION
1691 023600  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     023604  000404                        BR      9$              ;GO TO 9$ IF NO ERROR
     023606  000240                        NOP                     ;RETURN HERE IF ERROR
     023610  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     023612  000137  025004                JMP     42$             ;GO TO 42$ IF ERROR
1692 023616                             9$:
1693
1694                                    ;VERIFY NO ERRORS DURING FORMAT
1695 023616  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     023622  000405                        BR      10$             ;GO TO 10$ IF NO ERROR
     023624  000240                        NOP                     ;RETURN HERE IF ERROR
     023626  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     023630  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023632  000137  025004                JMP     42$             ;GO TO 42$ IF ERROR
1696 023636                             10$:
1697
1698                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1699 023636  012737  023646  001124            MOV     #11$,$LPERR
1700 023644  000410                        BR      12$             ;SKIP TO WRITE OPERATION
1701
1702                                    ;;*****************************************************************
1703                                    ;LOOP #2        WRITE,READ
1704
1705 023646                             11$:
1706
1707                                    ;PREPARE DEVICE FOR WRITE OPERATION
1708 023646  004737  037316                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     023652  154130                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                    ;VERIFY RECALIBRATION
     023654  000404                        BR      12$             ;GO TO 12$ IF NO ERROR
     023656  000240                        NOP                     ;RETURN HERE IF ERROR
     023660  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     023662  000137  025004                JMP     42$             ;GO TO 42$ IF ERROR
1709 023666                             12$:
1710
1711                                    ;WRITE DATA TO THE DRIVE
1712 023666                             13$:
1713 023666  012737  106442  001416            MOV     #BUFONE+4,RMBAO ;CHANGE BUS ADDRESS
1714 023674  012737  177400  001414            MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
1715 023702  012737  000061  001412            MOV     #WD.GO,RMCS10   ;WRITE DATA COMMAND
1716 023710  012702  001553                    MOV     #PUTINX,R2      ;LOAD PUT REGISTER INDEX TABLE
1717 023714  112722  000006                    MOVB    #RMDA,(R2)+
1718 023720  112722  000034                    MOVB    #RMDC,(R2)+
```

```
 1719 023724  112722  000032                  MOVB    #RMOF,(R2)+
 1720 023730  112722  000004                  MOVB    #RMBA,(R2)+
 1721 023734  112722  000002                  MOVB    #RMWC,(R2)+
 1722 023740  112722  000000                  MOVB    #RMCS1,(R2)+
 1723 023744  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
 1724
 1725 023750  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      023754  000404                          BR      14$             ;GO TO 14$ IF NO ERROR
      023756  000240                          NOP                     ;RETURN HERE IF ERROR
      023760  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      023762  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
 1726 023766                          14$:
 1727
 1728                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      023766  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 1729
 1730                                 ;WAIT FOR COMMAND TO COMPLETE
      023772  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 1731
 1732                                 ;GO READ STATUS FOR WRITE COMMAND
 1733 023776  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      024002  000404                          BR      15$             ;GO TO 15$ IF NO ERROR
      024004  000240                          NOP                     ;RETURN HERE IF ERROR
      024006  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      024010  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
 1734 024014                          15$:
 1735
 1736                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
 1737 024014  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      024020  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
      024022  000240                          NOP                     ;RETURN HERE IF ERROR
      024024  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      024026  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      024030  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
 1738
 1739                                 ;DECODE THE TYPE OF ERROR PRODUCED BY THE BIT POSITION OF R3
 1740 024034                          16$:
 1741 024034  005704                          TST     R4              ;IS THIS THE FIRST HEADER WORD ?
 1742 024036  001006                          BNE     17$             ;NO !!
 1743
 1744 024040  032703  010000                  BIT     #FMT16,R3       ;SHOULD FER BE SET ?
 1745 024044  001037                          BNE     20$             ;YES !.
 1746 024046  032703  140000                  BIT     #MSE.USE,R3     ;SHOULD BSE BE SET ?
 1747 024052  001061                          BNE     22$             ;YES !.
 1748
 1749                                 ;VERIFY THAT A HEADER COMPARE ERROR WAS DETECTED
 1750 024054                          17$:
 1751 024054  032737  000200  001352          BIT     #HCE,RMER1I     ;WAS HCE DETECTED ??
 1752 024062  001102                          BNE     24$             ;YES !!
 1753
 1754 024064  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      024070  000405                          BR      18$             ;GO TO 18$ IF NO ERROR
      024072  000240                          NOP                     ;RETURN HERE IF ERROR
      024074  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      024076  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      024100  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
 1755 024104                          18$:
```

```
1756 024104  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1757 024112  012737  000200  001140          MOV     #HCE,$GDDAT     ;EXPECTED STATUS
1758 024120  010437  001174                  MOV     R4,$TMP0        ;R4 = HEADER WORD NUMBER AND
1759 024124  001002                          BNE     19$             ;ADJUST NUMBER.
1760 024126  005237  001174                  INC     $TMP0
1761 024132  010337  001176          19$:    MOV     R3,$TMP1        ;R3 = BIT POSITION
1762 024136  104344                          EMT     344
1763 024140  000137  025004                  JMP     42$
1764
1765                                 ;VERIFY THAT A FORMAT ERROR WAS DETECTED
1766 024144                          20$:
1767 024144  032737  000020  001352          BIT     #FER,RMER1I     ;WAS FER DETECTED ??
1768 024152  001046                          BNE     24$             ;YES .!
1769
1770 024154  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024160  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
     024162  000240                          NOP                     ;RETURN HERE IF ERROR
     024164  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024166  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024170  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
1771 024174                          21$:
1772 024174  012737  000020  001140          MOV     #FER,$GDDAT     ;EXPECTED STATUS
1773 024202  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1774 024210  104343                          EMT     343
1775 024212  000137  025004                  JMP     42$
1776
1777                                 ;VERIFY THAT A BAD SECTOR ERROR WAS DETECTED
1778 024216                          22$:
1779 024216  032737  100000  001400          BIT     #BSE,RMER2I     ;WAS BSE DETECTED ??
1780 024224  001021                          BNE     24$             ;YES !!
1781
1782 024226  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024232  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
     024234  000240                          NOP                     ;RETURN HERE IF ERROR
     024236  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024240  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024242  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
1783 024246                          23$.
1784 024246  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
1785 024254  012737  100000  001140          MOV     #BSE,$GDDAT     ;EXPECTED STAIUS
1786 024262  104345                          EMT     345
1787 024264  000137  025004                  JMP     42$
1788
1789                                 ;CHECK FOR OTHER ERRORS
1790 024270                          24$:
1791 024270  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     024274  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
     024276  000240                          NOP                     ;RETURN HERE IF ERROR
     024300  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     024302  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024304  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
1792 024310                          25$:
1793
1794                                 ;CHANGE LOOP ADDRESSES
1795 024310  012737  024316  001124          MOV     #26$,$LPERR
1796
1797                                 ;;*****************************************************************
```

```
                                        ;LOOP #3        READ
                                        26$:
                                        ;PREPARE DEVICE FOR READ OPERATION
024316  004737  037316                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
024322  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                ;VERIFY RECALIBRATION
024324  000404                          BR      27$             ;GO TO 27$ IF NO ERROR
024326  000240                          NOP                     ;RETURN HERE IF ERROR
024330  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
024332  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
024336                                  27$:

                                        ;READ DATA FROM DEVICE
024336                                  28$:
024336  012737  107446  001416                  MOV     #BUFTWO+4,RMBAO ;CHANGE BUS ADDRESS
024344  012737  177400  001414                  MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
024352  012737  000071  001412                  MOV     #RD!GO,RMCS10   ;READ DATA COMMAND
024360  012702  001553                          MOV     #PUTINX,R2      ;LOAD PUT REGISTER INDEX TABLE
024364  112722  000006                          MOVB    #RMDA,(R2)+
024370  112722  000032                          MOVB    #RMOF,(R2)+
024374  112722  000034                          MOVB    #RMDC,(R2)+
024400  112722  000004                          MOVB    #RMBA,(R2)+
024404  112722  000002                          MOVB    #RMWC,(R2)+
024410  112722  000000                          MOVB    #RMCS1,(R2)+
024414  112712  000200                          MOVB    #200,(R2)

024420  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
024424  000404                          BR      29$             ;GO TO 29$ IF NO ERROR
024426  000240                          NOP                     ;RETURN HERE IF ERROR
024430  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
024432  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
024436                                  29$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
024436  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
024442  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR READ OPERATION
024446  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
024452  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
024454  000240                          NOP                     ;RETURN HERE IF ERROR
024456  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
024460  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
024464                                  30$:

                                        ;CHECK FOR ERRORS DURING READ OPERATION
024464  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
```

```
        024470  000405                          BR      31$             ;GO TO 31$ IF NO ERROR
        024472  000240                          NOP                     ;RETURN HERE IF ERROR
        024474  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        024476  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024500  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
1833    024504                          31$:
1834    024504  005704                          TST     R4              ;IS THIS THE FIRST HEADER WORD ?
1835    024506  001006                          BNE     32$             ;NO !!
1836
1837    024510  032703  010000                  BIT     #FMT16,R3       ;SHOULD FER BE SET ?
1838    024514  001037                          BNE     35$             ;YES !!
1839    024516  032703  140000                  BIT     #MSE.USE,R3     ;SHOULD BSE BE SET ?
1840    024522  001061                          BNE     37$             ;YES !!
1841
1842                                    ;VERIFY THAT A HEADER COMPARE ERROR WAS DETECTED
1843    024524                          32$:
1844    024524  032737  000200  001352          BIT     #HCE,RMER1I     ;WAS HCE DETECTED ??
1845    024532  001102                          BNE     39$             ;YES !!
1846
1847    024534  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        024540  000405                          BR      33$             ;GO TO 33$ IF NO ERROR
        024542  000240                          NOP                     ;RETURN HERE IF ERROR
        024544  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        024546  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024550  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
1848    024554                          33$:
1849    024554  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1850    024562  012737  000200  001140          MOV     #HCE,$GDDAT     ;EXPECTED STATUS
1851    024570  010437  001174                  MOV     R4,$TMP0        ;R4 = HEADER WORD NUMBER AND
1852    024574  001002                          BNE     34$             ;ADJUST NUMBER.
1853    024576  005237  001174                  INC     $TMP0
1854    024602  010337  001176          34$:    MOV     R3,$TMP1        ;R3 = BIT POSITION
1855    024606  104344                          EMT     344
1856    024610  000137  025004                  JMP     42$
1857
1858                                    ;VERIFY THAT A FORMAT ERROR WAS DETECTED
1859    024614                          35$:
1860    024614  032737  000020  001352          BIT     #FER,RMER1I     ;WAS FER DETECTED ??
1861    024622  001046                          BNE     39$             ;YES !!
1862
1863    024624  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        024630  000405                          BR      36$             ;GO TO 36$ IF NO ERROR
        024632  000240                          NOP                     ;RETURN HERE IF ERROR
        024634  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        024636  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024640  000137  025004                  JMP     42$             ;GO TO 42$ IF ERROR
1864    024644                          36$:
1865    024644  012737  000020  00[illegible]    MOV     #FER,$GDDAT     ;EXPECTED STATUS
1866    024652  013737  001352  00114[illegible] MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1867    024660  104343                          EMT     343
1868    024662  000137  025004                  JMP     42$
1869
1870                                    ;VERIFY THAT A BAD SECTOR ERROR WAS DETECTED
1871    024666                          37$:
1872    024666  032737  100000  001400          BIT     #BSE,RMER2I     ;WAS BSE DETECTED ??
1873    024674  001021                          BNE     39$             ;YES ..
1874
```

```
1875 024676  004737  056572                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024702  000405                        BR      38$             ;GO TO 38$ IF NO ERROR
     024704  000240                        NOP                     ;RETURN HERE IF ERROR
     024706  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024710  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024712  000137  025004                JMP     42$             ;GO TO 42$ IF ERROR
1876 024716                          38$:
1877 024716  013737  001400  001142        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
1878 024724  012737  100000  001140        MOV     #BSE,$GDDAT     ;EXPECTED STAIUS
1879 024732  104345                        EMT     345
1880 024734  000137  000350                JMP     350
1881
1882                                 ;CHECK FOR OTHER ERRORS
1883 024740                          39$:
1884 024740  004737  044710                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     024744  000405                        BR      40$             ;GO TO 40$ IF NO ERROR
     024746  000240                        NOP                     ;RETURN HERE IF ERROR
     024750  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     024752  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024754  000137  025004                JMP     42$             ;GO TO 42$ IF ERROR
1885 024760                          40$:
1886 024760  006303                        ASL     R3              ;SHIFT TO NEXT BIT
1887 024762  001006                        BNE     41$             ;REPEAT IF NOT DONE
1888
1889 024764  005704                        TST     R4              ;SECOND HEADER WORD DONE??
1890 024766  001006                        BNE     42$             ;YES!!
1891 024770  012704  000002                MOV     #2,R4           ;SETUP FOR 2ND HEADER WORD
1892 024774  012703  000001                MOV     #1,R3           ;SETUP FOR HCE
1893 025000                          41$:
1894 025000  000137  023346                JMP     2$
1895 025004                          42$:
1896 025004  000464                        BR      46$             ;EXIT IF ERROR
1897                                 ;;*****************************************************************
1898                                 ;*REFORMAT SECTOR THAT WAS WRITTEN WITH BAD HEADER
1899                                 ;;*****************************************************************
1900 025006                          43$:
1901 025006  012737  177776  001414        MOV     #-2,RMWCO               ;ONLY TWO HEAD WORDS
1902 025014  012737  010000  001444        MOV     #FMT16,RMOFO            ;ALWAYS IN 16 BITS MODE
1903 025022  012737  106436  001416        MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS,REFORMAT THE SECTOR
1904 025030  012737  000062  001412        MOV     #WH,RMCS1O              ;WRITE HEAD AND DATA COMMAND
1905 025036  004737  042174                JSR     PC,GENBUF               ;SET UP THE BUFFER
1906
1907 025042  004737  037316                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     025046  054130                        .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                    ;VERIFY RECALIBRATION
     025050  000404                        BR      44$             ;GO TO 44$ IF NO ERROR
     025052  000240                        NOP                     ;RETURN HERE IF ERROR
     025054  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     025056  000137  025062                JMP     44$             ;GO TO 44$ IF ERROR
1908 025062                          44$:
1909 025062  012737  000063  001412        MOV     #WH!GO,RMCS1O           ;FORMAT THE SECTOR
1910 025070  012702  001553                MOV     #PUTINX,R2              ;SET UP THE REGS
```

```
 1911 025074  112722  000006                MOVB    #RMDA,(R2)+
 1912 025100  112722  000032                MOVB    #RMOF,(R2)+
 1913 025104  112722  000034                MOVB    #RMDC,(R2)+
 1914 025110  112722  000004                MOVB    #RMBA,(R2)+
 1915 025114  112722  000002                MOVB    #RMWC,(R2)+
 1916 025120  112722  000000                MOVB    #RMCS1,(R2)+
 1917 025124  112722  000200                MOVB    #200,(R2)+
 1918 025130  004737  043330                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      025134  000404                        BR      45$             ;GO TO 45$ IF NO ERROR
      025136  000240                        NOP                     ;RETURN HERE IF ERROR
      025140  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      025142  000137  025156                JMP     46$             ;GO TO 46$ IF ERROR
 1919 025146  000240              45$:     NOP
 1920 025150  004737  043672                JSR     PC,TIMOUT               ;WAIT FOR FINISH
 1921 025154  000240                        NOP
 1922 025156                      46$:
 1923
 1924                             ;;*****************************************************************
                                  ;*TEST 15       WRITE, READ W/ HCI
                                  ;;*****************************************************************
      025156                      TST15:
      025156  000004                      SCOPE                   ;SCOPE CALL
      025160  000240                      NOP                     ;START OF TEST
      025162  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      025166  013700  001276              MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      025172  013701  001466              MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      025176  012737  000015  001226      MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1925
 1926                             ;;*****************************************************************
 1927                             ;LOOP #1       FORMAT,WRITE,READ
 1928
 1929 025204  012704  000000      1$:     MOV     #0,R4           ;HEADER WORD
 1930 025210  012703  000001              MOV     #1,R3           ;HCE BIT
 1931 025214                      2$:
 1932
 1933                             ;PREPARE THE DEVICE FOR FORMAT OPERATION
 1934 025214  004737  037316              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      025220  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                  ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                  ;CLEAR CONTROLLER & SELECT DEVICE
                                                                  ;VERIFY CONTROLLER CLEAR OPERATION
                                                                  ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                  ;VERIFY PACK ACKNOWLEDGE
                                                                  ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                  ;VERIFY RECALIBRATION
      025222  000404                      BR      3$              ;GO TO 3$ IF NO ERROR
      025224  000240                      NOP                     ;RETURN HERE IF ERROR
      025226  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      025230  000137  026562              JMP     41$             ;GO TO 41$ IF ERROR
 1935
 1936                             ;LOAD PARAMETERS AND GENERATE DATA BUFFER
 1937 025234                      3$:
 1938 025234  012737  000000  001446      MOV     #0,RMDCO        ;CYLINDER = 0
 1939 025242  012737  000000  001420      MOV     #0,RMDAO        ;TRACK = 0, SECTOR - 0
 1940 025250  012737  106436  001416      MOV     #BUFONE,RMBAO   ;BUS ADDRESS
 1941 025256  012737  177376  001414      MOV     #-258.,RMWCO    ;2 + 256. WORDS (2' COMP)
 1942 025264  012737  010000  001444      MOV     #FMT16,RMOFO    ;16 BIT FORMAT
```

```
1943 025272  012737  000063  001412          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA COMMAND
1944
1945                                  ;VERIFY THAT SECTOR IS NOT BAD
     025300  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     025304  000405                          BR      4$              ;GO TO 4$ IF NO ERROR
     025306  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     025312  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     025314  000137  026410                  JMP     37$             ;GO TO 37$ IF ERROR
1946 025320                           4$:
1947 025320  012737  072016  001174          MOV     #ONES,$TMP0     ;STARTING ADDRESS OF PATTERN
1948 025326  012737  000001  001176          MOV     #1,$TMP1        ;RANGE OF PATTERN
1949 025334  004737  042174                  JSR     PC,GENBUF       ;GO GENERATE BUFFER FOR FORMAT
1950
1951                                  ;CHANGE HEADER WORD TO FORCE ERROR DURING WRITE
1952 025340  030364  106436                  BIT     R3,BUFONE(R4)   ;SET OR RESET BIT??
1953 025344  001403                          BEQ     5$
1954 025346  040364  106436                  BIC     R3,BUFONE(R4)   ;RESET BIT
1955 025352  000402                          BR      6$
1956 025354  050364  106436           5$:    BIS     R3,BUFONE(R4)   ;SET BIT
1957
1958                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
1959 025360                           6$:
1960 025360  012702  001553                  MOV     #PUTINX,R2      ;R2 = BYTE ENTRY POSITION
1961 025364  112722  000034                  MOVB    #RMDC,(R2)+
1962 025370  112722  000006                  MOVB    #RMDA,(R2)+
1963 025374  112722  000004                  MOVB    #RMBA,(R2)+
1964 025400  112722  000002                  MOVB    #RMWC,(R2)+
1965 025404  112722  000032                  MOVB    #RMOF,(R2)+
1966 025410  112722  000000                  MOVB    #RMCS1,(R2)+
1967 025414  112712  000200                  MOVB    #200,(R2)       ;TERMINATE TABLE
1968 025420                           7$:
1969
1970                                  ;FORMAT THE DRIVE
1971 025420  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     025424  000404                          BR      8$              ;GO TO 8$ IF NO ERROR
     025426  000240                          NOP                     ;RETURN HERE IF ERROR
     025430  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     025432  000137  026410                  JMP     37$             ;GO TO 37$ IF ERROR
1972 025436                           8$:
1973
1974                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     025436  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1975
1976                                  ;WAIT FOR COMMAND TO COMPLETE
     025442  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1977
1978                                  ;GO READ STATUS FOR FORMAT OPERATION
1979 025446  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     025452  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
     025454  000240                          NOP                     ;RETURN HERE IF ERROR
     025456  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     025460  000137  026410                  JMP     37$             ;GO TO 37$ IF ERROR
1980 025464                           9$:
1981
1982                                  ;VERIFY NO ERRORS DURING FORMAT
1983 025464  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     025470  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
```

```
        025472  000240                          NOP                     ;RETURN HERE IF ERROR
        025474  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        025476  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        025500  000137  026410                  JMP     37$             ;GO TO 37$ IF ERROR
1984    025504                          10$:
1985
1986                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1987    025504  012737  025514  001124          MOV     #11$,$LPERR
1988    025512  000410                          BR      12$             ;SKIP TO WRITE OPERATION
1989
1990                                    ;;*****************************************************************
1991                                    ;LOOP #2        WRITE,READ
1992
1993    025514                          11$:
1994
1995                                    ;PREPARE DEVICE FOR WRITE OPERATION
1996    025514  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        025520  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        025522  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
        025524  000240                          NOP                     ;RETURN HERE IF ERROR
        025526  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        025530  000137  026410                  JMP     37$             ;GO TO 37$ IF ERROR
1997    025534                          12$:
1998
1999                                    ;WRITE DATA TO THE DRIVE
2000    025534                          13$:
2001    025534  012737  012000  001444          MOV     #FMT16!HCI,RMOFO        ;INHIBIT HEADER COMPARE
2002    025542  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2003    025550  012737  106442  001416          MOV     #BUFONE+4,RMBAO         ;LOAD STARTING BUFFER ADDRESS
2004    025556  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2005    025562  112722  000006                  MOVB    #RMDA,(R2)+
2006    025566  112722  000034                  MOVB    #RMDC,(R2)+
2007    025572  112722  000032                  MOVB    #RMOF,(R2)+
2008    025576  112722  000004                  MOVB    #RMBA,(R2)+
2009    025602  112722  000002                  MOVB    #RMWC,(R2)+
2010    025606  112722  000000                  MOVB    #RMCS1,(R2)+
2011    025612  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
2012
2013    025616  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        025622  000404                          BR      14$             ;GO TO 14$ IF NO ERROR
        025624  000240                          NOP                     ;RETURN HERE IF ERROR
        025626  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        025630  000137  026410                  JMP     37$             ;GO TO 37$ IF ERROR
2014    025634                          14$:
2015
2016                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        025634  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2017
2018                                    ;WAIT FOR COMMAND TO COMPLETE
        025640  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
```

```
                                        ;GO READ STATUS FOR WRITE COMMAND
025644  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
025650  000404                                  BR      15$             ;GO TO 15$ IF NO ERROR
025652  000240                                  NOP                     ;RETURN HERE IF ERROR
025654  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
025656  000137  026410                          JMP     37$             ;GO TO 37$ IF ERROR
025662                                  15$:
                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
025662  004737  044056                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
025666  000405                                  BR      16$             ;GO TO 16$ IF NO ERROR
025670  000240                                  NOP                     ;RETURN HERE IF ERROR
025672  104000                                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
025674  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
025676  000137  026410                          JMP     37$             ;GO TO 37$ IF ERROR
025702  005037  001140          16$:    CLR     $GDDAT                  ;EXPECTED STATUS

025706  032737  000220  001352                  BIT     #HCE!FER,RMER1I         ;ANY ERROR?:
025714  001407                                  BEQ     17$
025716  013737  001352  001142                  MOV     RMER1I,$BDDAT           ;RECEIVED STATUS
025724  042737  177557  001142                  BIC     #^C<HCE!FER>,$BDDAT     ;CLEAR DONT CARES
025732  000412                                  BR      18$
025734                                  17$:
025734  013737  001400  001142                  MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
025742  032737  100000  001400                  BIT     #BSE,RMER2I             ;ANY BAD SECTOR ERROR ?
025750  001406                                  BEQ     19$                     ;NO !!
025752  042737  077777  001142                  BIC     #^CBSE,$BDDAT           ;SAVE BSE FOR ERROR
025760                                  18$:
025760  104346                                  EMT     346
025762  000137  026410                          JMP     37$
025766                                  19$:

025766                                  20$:
025766  004737  056572                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
025772  000405                                  BR      21$             ;GO TO 21$ IF NO ERROR
025774  000240                                  NOP                     ;RETURN HERE IF ERROR
025776  104000                                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
026000  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
026002  000137  026410                          JMP     37$             ;GO TO 37$ IF ERROR
026006                                  21$:
026006  004737  044710                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
026012  000405                                  BR      22$             ;GO TO 22$ IF NO ERROR
026014  000240                                  NOP                     ;RETURN HERE IF ERROR
026016  104000                                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
026020  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
026022  000137  026410                          JMP     37$             ;GO TO 37$ IF ERROR
026026                                  22$:

                                        ;CHANGE LOOP ADDRESSES
026026  012737  026036  001124                  MOV     #23$,$LPERR
026034  000410                                  BR      24$

                                        ;;*****************************************************************
                                        ;LOOP #3        READ

026036                                  23$:
```

```
                                        ;PREPARE DEVICE FOR READ OPERATION
026036  004737  037316                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
026042  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                      ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                      ;CLEAR CONTROLLER & SELECT DEVICE
                                                                      ;VERIFY CONTROLLER CLEAR OPERATION
                                                                      ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                      ;VERIFY PACK ACKNOWLEDGE
                                                                      ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                      ;VERIFY RECALIBRATION
026044  000404                                BR      24$             ;GO TO 24$ IF NO ERROR
026046  000240                                NOP                     ;RETURN HERE IF ERROR
026050  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
026052  000137  026410                        JMP     37$             ;GO TO 37$ IF ERROR
026056                                  24$:

                                        ;READ DATA FROM DEVICE
026056                                  25$:
026056  012737  000071  001412                MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
026064  012737  107446  001416                MOV     #BUFTWO+4,RMBA0         ;LOAD STARTING BUFFER ADDRESS
026072  012702  001553                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
026076  112722  000006                        MOVB    #RMDA,(R2)+
026102  112722  000032                        MOVB    #RMOF,(R2)+
026106  112722  000034                        MOVB    #RMDC,(R2)+
026112  112722  000004                        MOVB    #RMBA,(R2)+
026116  112722  000002                        MOVB    #RMWC,(R2)+
026122  112722  000000                        MOVB    #RMCS1,(R2)+
026126  112712  000200                        MOVB    #200,(R2)

026132  004737  043330                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
026136  000404                                BR      26$             ;GO TO 26$ IF NO ERROR
026140  000240                                NOP                     ;RETURN HERE IF ERROR
026142  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
026144  000137  026410                        JMP     37$             ;GO TO 37$ IF ERROR
026150                                  26$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
026150  004737  042774                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
026154  004737  043672                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR READ OPERATION
026160  004737  043060                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
026164  000404                                BR      27$             ;GO TO 27$ IF NO ERROR
026166  000240                                NOP                     ;RETURN HERE IF ERROR
026170  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
026172  000137  026410                        JMP     37$             ;GO TO 37$ IF ERROR
026176                                  27$:

                                        ;CHECK FOR ERRORS DURING READ OPERATION
026176  004737  044056                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
026202  000405                                BR      28$             ;GO TO 28$ IF NO ERROR
026204  000240                                NOP                     ;RETURN HERE IF ERROR
026206  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
026210  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
      026212  000137  026410                    JMP     37$             ;GO TO 37$ IF ERROR
2088  026216  005037  001140          28$:      CLR     $GDDAT                  ;EXPECTED STATUS
2089
2090  026222  032737  000220  001352            BIT     #HCE!FER,RMER1I         ;ANY ERROR ?
2091  026230  001407                            BEQ     29$                     ;NO!!
2092  026232  013737  001352  001142            MOV     RMER1I,$BDDAT           ;RECEIVED STATUS
2093  026240  042737  177557  001142            BIC     #^C<HCE.FER>,$BDDAT     ;CLEAR DONT CARES
2094  026246  000412                            BR      30$
2095  026250                          29$:
2096  026250  013737  001400  001142            MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
2097  026256  032737  100000  001400            BIT     #BSE,RMER2I             ;ANY BAD SECTOR ERROR ?
2098  026264  001406                            BEQ     31$                     ;NO !.
2099  026266  042737  077777  001142            BIC     #^CBSE,$BDDAT           ;CLEAR DONT CARES
2100  026274                          30$:
2101  026274  104346                            EMT     346
2102  026276  000137  026410                    JMP     37$
2103  026302                          31$:
2104
2105  026302                          32$:
2106  026302  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      026306  000405                            BR      33$             ;GO TO 33$ IF NO ERROR
      026310  000240                            NOP                     ;RETURN HERE IF ERROR
      026312  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      026314  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026316  000137  026410                    JMP     37$             ;GO TO 37$ IF ERROR
2107  026322                          33$:
2108  026322  004737  044710                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      026326  000405                            BR      34$             ;GO TO 34$ IF NO ERROR
      026330  000240                            NOP                     ;RETURN HERE IF ERROR
      026332  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      026334  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026336  000137  026410                    JMP     37$             ;GO TO 37$ IF ERROR
2109  026342                          34$:
2110  026342  004737  042432                    JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
      026346  106442                            .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
      026350  107446                            .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
      026352  000404                            BR      35$             ;GO TO 35$ IF NO ERROR
      026354  000240                            NOP                     ;RETURN HERE IF ERROR
      026356  104000                            EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
      026360  000137  026410                    JMP     37$             ;GO TO 37$ IF ERROR
2111  026364                          35$:
2112  026364  006303                            ASL     R3                      ;SHIFT HCE BIT
2113  026366  001006                            BNE     36$                     ;CONTINUE IF NOT DONE
2114  026370  005704                            TST     R4                      ;SECOND HEADER DONE??
2115  026372  001006                            BNE     37$                     ;YES.!
2116  026374  012703  000001                    MOV     #1,R3                   ;START WITH BIT 0
2117  026400  012704  000002                    MOV     #2,R4                   ;DO SECOND HEADER WORD
2118  026404                          36$:
2119  026404  000137  025214                    JMP     2$
2120  026410                          37$:
2121  026410  000464                            BR      41$             ;EXIT IF ERROR
2122                                  ;;*****************************************************************
2123                                  ;*REFORMAT SECTOR THAT WAS WRITTEN WITH BAD HEADER
2124                                  ;;*****************************************************************
2125  026412                          38$:
2126  026412  012737  177776  001414            MOV     #-2,RMWCO               ;WORD COUNT
2127  026420  012737  010000  001444            MOV     #FMT16,RMOFO            ;IN 16 BIT MODE
```

```
 2128 026426  012737  106436  001416          MOV     #BUFONE,RMBA0           ;BUFFER ADDRESS
 2129 026434  012737  000062  001412          MOV     #WH,RMCS10              ;FORMAT COMMAND
 2130 026442  004737  042174                  JSR     PC,GENBUF               ;SET UP BUFFER
 2131
 2132 026446  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      026452  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                      ;CLEAR CONTROLLER & SELECT DEVICE
                                                                      ;VERIFY CONTROLLER CLEAR OPERATION
                                                                      ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                      ;VERIFY PACK ACKNOWLEDGE
                                                                      ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                      ;VERIFY RECALIBRATION
      026454  000404                          BR      39$             ;GO TO 39$ IF NO ERROR
      026456  000240                          NOP                     ;RETURN HERE IF ERROR
      026460  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      026462  000137  026466                  JMP     39$             ;GO TO 39$ IF ERROR
 2133 026466                          39$:
 2134 026466  012737  000063  001412          MOV     #WH!GO,RMCS10
 2135 026474  012702  001553                  MOV     #PUTINX,R2              ;TABLE ADDRESS
 2136 026500  112722  000006                  MOVB    #RMDA,(R2)+
 2137 026504  112722  000032                  MOVB    #RMOF,(R2)+
 2138 026510  112722  000034                  MOVB    #RMDC,(R2)+
 2139 026514  112722  000004                  MOVB    #RMBA,(R2)+
 2140 026520  112722  000002                  MOVB    #RMWC,(R2)+
 2141 026524  112722  000000                  MOVB    #RMCS1,(R2)+
 2142 026530  112722  000200                  MOVB    #200,(R2)+
 2143 026534  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      026540  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
      026542  000240                          NOP                     ;RETURN HERE IF ERROR
      026544  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      026546  000137  026562                  JMP     41$             ;GO TO 41$ IF ERROR
 2144 026552  000240                  40$:    NOP
 2145 026554  004737  043672                  JSR     PC,TIMOUT
 2146 026560  000240                          NOP
 2147 026562                          41$:
 2148
 2149                                         ;;*******************************************************************
                                              ;*TEST 16        WRITE, READ W/ IVC ERROR
                                              ;;*******************************************************************
      026562                          TST16:
      026562  000004                          SCOPE                   ;SCOPE CALL
      026564  000240                          NOP                     ;START OF TEST
      026566  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      026572  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      026576  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      026602  012737  000016  001226          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2150
 2151                                         ;;*******************************************************************
 2152                                         ;LOOP #1        WRITE,READ
 2153
 2154                                         ;LOAD PARAMETERS AND GENERATE DATA BUFFER
 2155 026610                          1$:
 2156 026610  012737  000000  001446          MOV     #0,RMDC0        ;CYLINDER = 0
 2157 026616  012737  000000  001420          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
 2158 026624  012737  106436  001416          MOV     #BUFONE,RMBA0   ;BUS ADDRESS
 2159 026632  012737  177400  001414          MOV     #-256.,RMWC0    ;256. WORDS (2'S COMP)
 2160 026640  012737  010000  001444          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
```

```
2161 026646  012737  000060  001412          MOV     #WD,RMCS1O      ;WRITE HEADER AND DATA COMMAND
2162
2163                                  ;VERIFY THAT SECTOR IS NOT BAD
     026654  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     026660  000405                          BR      2$              ;GO TO 2$ IF NO ERROR
     026662  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     026666  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     026670  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
2164 026674                           2$:
2165 026674  012737  001420  001174          MOV     #RMDAO,$TMPO    ;USE SECTOR FOR DATA PATTERN
2166 026702  012737  000001  001176          MOV     #1,$TMP1
2167 026710  004737  042174                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
2168
2169                                  ;PREPARE DEVICE FOR WRITE OPERATION
2170 026714  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     026720  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                     ;VERIFY RECALIBRATION
     026722  000404                          BR      3$              ;GO TO 3$ IF NO ERROR
     026724  000240                          NOP                     ;RETURN HERE IF ERROR
     026726  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     026730  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
2171 026734                           3$:
2172
2173                                  ;RESET VOLUME VALID
2174 026734  112737  000024  001553          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     026742  112737  000200  001554          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026750  012737  000001  001436          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER - DMD
     026756  004737  043330                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     026762  000404                          BR      4$              ;GO TO 4$ IF NO ERROR
     026764  000240                          NOP                     ;RETURN HERE IF ERROR
     026766  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026770  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
2175 026774                           4$:
2176
2177                                  ;SETUP PARAMETERS AND EXCUTE WRITE DATA COMMAND
2178 026774                           5$:
2179 026774  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2180 027002  012737  000000  001436          MOV     #0,RMMR10               ;RESET DIAGNOSTIC MODE
2181 027010  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2182 027014  112722  000024                  MOVB    #RMMR1,(R2)+
2183 027020  112722  000006                  MOVB    #RMDA,(R2)+
2184 027024  112722  000034                  MOVB    #RMDC,(R2)+
2185 027030  112722  000032                  MOVB    #RMOF,(R2)+
2186 027034  112722  000002                  MOVB    #RMWC,(R2)+
2187 027040  112722  000004                  MOVB    #RMBA,(R2)+
2188 027044  112722  000000                  MOVB    #RMCS1,(R2)+
2189 027050  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
2190
2191 027054  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027060  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
     027062  000240                          NOP                     ;RETURN HERF IF ERROR
```

```
        027064  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        027066  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
 2192   027072                          6$:
 2193
 2194                                   ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        027072  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2195
 2196                                   ;WAIT FOR COMMAND TO COMPLETE
        027076  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2197
 2198                                   ;GO READ STATUS FOR WRITE COMMAND
 2199   027102  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        027106  000404                          BR      7$              ;GO TO 7$ IF NO ERROR
        027110  000240                          NOP                     ;RETURN HERE IF ERROR
        027112  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        027114  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
 2200   027120                          7$:
 2201
 2202                                   ;CHECK FOR ERRORS DURING WRITE OPERATION
 2203   027120  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        027124  000405                          BR      8$              ;GO TO 8$ IF NO ERROR
        027126  000240                          NOP                     ;RETURN HERE IF ERROR
        027130  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        027132  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        027134  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
 2204   027140                          8$:
 2205   027140  032737  010000  001400          BIT     #IVC,RMER2I             ;WAS ''IVC'' DETECTED??
 2206   027146  001024                          BNE     10$                     ;YES!.
 2207
 2208   027150  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        027154  000405                          BR      9$              ;GO TO 9$ IF NO ERROR
        027156  000240                          NOP                     ;RETURN HERE IF ERROR
        027160  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        027162  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        027164  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
 2209   027170                          9$:
 2210   027170  013737  001400  001142          MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
 2211   027176  013737  001400  001140          MOV     RMER2I,$GDDAT           ;EXPECTED STATUS
 2212   027204  052737  010000  001140          BIS     #IVC,$GDDAT
 2213   027212  104342                          EMT     342
 2214   027214  000137  027602                  JMP     22$
 2215   027220                          10$:
 2216   027220  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        027224  000405                          BR      11$             ;GO TO 11$ IF NO ERROR
        027226  000240                          NOP                     ;RETURN HERE IF ERROR
        027230  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        027232  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        027234  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
 2217   027240                          11$:
 2218
 2219                                   ;CHANGE LOOP ADDRESSES
 2220   027240  012737  027250  001124          MOV     #12$,$LPERR
 2221   027246  000410                          BR      13$
 2222
 2223                                   ;;*****************************************************************
 2224                                   ;LOOP #2        READ
 2225
```

```
2226 027250                          12$:
2227
2228                                 ;PREPARE DEVICE FOR READ OPERATION
2229 027250  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     027254  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                             ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                             ;CLEAR CONTROLLER & SELECT DEVICE
                                             ;VERIFY CONTROLLER CLEAR OPERATION
                                             ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                             ;VERIFY PACK ACKNOWLEDGE
                                             ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                             ;VERIFY RECALIBRATION
     027256  000404                          BR      13$             ;GO TO 13$ IF NO ERROR
     027260  000240                          NOP                     ;RETURN HERE IF ERROR
     027262  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     027264  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
2230 027270                          13$:
2231
2232                                 ;RESET VOLUME VALID
2233 027270  112737  000024  001553          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     027276  112737  000200  001554          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     027304  012737  000001  001436          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER - DMD
     027312  004737  043330                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     027316  000404                          BR      14$             ;GO TO 14$ IF NO ERROR
     027320  000240                          NOP                     ;RETURN HERE IF ERROR
     027322  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     027324  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
2234 027330                          14$:
2235
2236                                 ;READ DATA FROM DEVICE
2237 027330                          15$:
2238 027330  012737  000071  001412          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
2239 027336  012737  107442  001416          MOV     #BUFTWO,RMBA0           ;LOAD STARTING BUFFER ADDRESS
2240 027344  012737  000000  001436          MOV     #0,RMMR10               ;RESET DIANOSTIC MODE
2241 027352  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2242 027356  112722  000024                  MOVB    #RMMR1,(R2)+
2243 027362  112722  000006                  MOVB    #RMDA,(R2)+
2244 027366  112722  000032                  MOVB    #RMOF,(R2)+
2245 027372  112722  000034                  MOVB    #RMDC,(R2)+
2246 027376  112722  000004                  MOVB    #RMBA,(R2)+
2247 027402  112722  000002                  MOVB    #RMWC,(R2)+
2248 027406  112722  000000                  MOVB    #RMCS1,(R2)+
2249 027412  112712  000200                  MOVB    #200,(R2)
2250
2251 027416  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027422  000404                          BR      16$             ;GO TO 16$ IF NO ERROR
     027424  000240                          NOP                     ;RETURN HERE IF ERROR
     027426  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     027430  000137  027602                  JMP     22$             ;GO TO 22$ IF ERROR
2252 027434                          16$:
2253
2254                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     027434  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2255
2256                                 ;WAIT FOR COMMAND TO COMPLETE
     027440  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2257
```

```
2258                                    ;GO READ STATUS FOR READ OPERATION
2259 027444  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027450  000404                            BR      17$             ;GO TO 17$ IF NO ERROR
     027452  000240                            NOP                     ;RETURN HERE IF ERROR
     027454  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027456  000137  027602                    JMP     22$             ;GO TO 22$ IF ERROR
2260 027462                             17$:
2261
2262                                    ;CHECK FOR ERRORS DURING READ OPERATION
2263 027462  004737  044056                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     027466  000405                            BR      18$             ;GO TO 18$ IF NO ERROR
     027470  000240                            NOP                     ;RETURN HERE IF ERROR
     027472  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     027474  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027476  000137  027602                    JMP     22$             ;GO TO 22$ IF ERROR
2264 027502                             18$:
2265 027502  032737  010000  001400            BIT     #IVC,RMER2I             ;WAS "IVC" DETECTED??
2266 027510  001024                            BNE     20$                     ;YES!!
2267
2268 027512  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     027516  000405                            BR      19$             ;GO TO 19$ IF NO ERROR
     027520  000240                            NOP                     ;RETURN HERE IF ERROR
     027522  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     027524  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027526  000137  027602                    JMP     22$             ;GO TO 22$ IF ERROR
2269 027532                             19$:
2270 027532  013737  001400  001140            MOV     RMER2I,$GDDAT           ;EXPECTED STATUS
2271 027540  052737  010000  001140            BIS     #IVC,$GDDAT
2272 027546  013737  001400  001142            MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
2273 027554  104342                            EMT     342
2274 027556  000137  027602                    JMP     22$
2275 027562                             20$:
2276 027562  004737  044710                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     027566  000405                            BR      21$             ;GO TO 21$ IF NO ERROR
     027570  000240                            NOP                     ;RETURN HERE IF ERROR
     027572  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     027574  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027576  000137  027602                    JMP     22$             ;GO TO 22$ IF ERROR
2277 027602                             21$:
2278
2279 027602                             22$:
2280
2281                                    ;;*******************************************************************
                                        ;*TEST 17        WRITE, READ W/ ABORT
                                        ;;*******************************************************************
     027602                             TST17:
     027602  000004                            SCOPE                   ;SCOPE CALL
     027604  000240                            NOP                     ;START OF TEST
     027606  012706  001100                    MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     027612  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     027616  013701  001466                    MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     027622  012737  000017  001226            MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2282
2283                                    ;;*******************************************************************
2284                                    ;LOOP #1        WRITE,READ
2285
2286                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
```

```
 2287 027630  012737  000000  001446          MOV     #0,RMDC0                ;CYLINDER = 0
 2288 027636  012737  000000  001420          MOV     #0,RMDA0                ;TRACK = 0, SECTOR = 0
 2289 027644  012737  106436  001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
 2290 027652  012737  177400  001414          MOV     #-256.,RMWC0            ;256. WORDS (2'S COMP)
 2291 027660  012737  010000  001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
 2292 027666  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2293 027674                              1$:
 2294
 2295                                     ;PREPARE DEVICE FOR WRITE OPERATION
 2296 027674  004737  037316                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      027700  154130                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                          ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                          ;CLEAR CONTROLLER & SELECT DEVICE
                                                                          ;VERIFY CONTROLLER CLEAR OPERATION
                                                                          ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                          ;VERIFY PACK ACKNOWLEDGE
                                                                          ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                          ;VERIFY RECALIBRATION
      027702  000404                              BR      2$              ;GO TO 2$ IF NO ERROR
      027704  000240                              NOP                     ;RETURN HERE IF ERROR
      027706  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      027710  000137  030520                      JMP     21$             ;GO TO 21$ IF ERROR
 2297 027714                              2$:
 2298
 2299                                     ;SET UNSAFE ERROR
 2300 027714  112737  000014  001553          MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
      027722  112737  000200  001554          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      027730  012737  040000  001426          MOV     #UNS,RMER10     ;SET RMER1 OUTPUT BUFFER = UNS
      027736  004737  043330                  JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
      027742  000404                          BR      3$              ;GO TO 3$ IF NO ERROR
      027744  000240                          NOP                     ;RETURN HERE IF ERROR
      027746  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      027750  000137  030520                  JMP     21$             ;GO TO 21$ IF ERROR
 2301 027754                              3$:
 2302
 2303                                     ;WRITE DATA TO THE DRIVE
 2304 027754                              4$:
 2305 027754  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2306 027762  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2307 027766  112722  000006                  MOVB    #RMDA,(R2)+
 2308 027772  112722  000034                  MOVB    #RMDC,(R2)+
 2309 027776  112722  000032                  MOVB    #RMOF,(R2)+
 2310 030002  112722  000004                  MOVB    #RMBA,(R2)+
 2311 030006  112722  000002                  MOVB    #RMWC,(R2)+
 2312 030012  112722  000000                  MOVB    #RMCS1,(R2)+
 2313 030016  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 2314
 2315 030022  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      030026  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
      030030  000240                          NOP                     ;RETURN HERE IF ERROR
      030032  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      030034  000137  030520                  JMP     21$             ;GO TO 21$ IF ERROR
 2316 030040                              5$:
 2317
 2318                                     ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      030040  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2319
```

```
 2320                                   ;GO GET REGISTER STATUS
 2321 030044  004737  043060            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      030050  000404                    BR      6$              ;GO TO 6$ IF NO ERROR
      030052  000240                    NOP                     ;RETURN HERE IF ERROR
      030054  104000                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      030056  000137  030520            JMP     21$             ;GO TO 21$ IF ERROR
 2322 030062                            6$:
 2323 030062  032737  020000  001350    BIT     #PIP,RMDSI              ;WAS COMMAND EXECUTED?
 2324 030070  001412                    BEQ     7$                      ;NO, GO WAIT
 2325 030072  013737  001350  001140    MOV     RMDSI,$GDDAT            ;EXPECTED STATUS
 2326 030100  042737  020000  001140    BIC     #PIP,$GDDAT
 2327 030106  013737  001350  001142    MOV     RMDSI,$BDDAT            ;RECEIVED STATUS
 2328 030114  104347                    EMT     347
 2329 030116                            7$:
 2330
 2331                                   ;WAIT FOR COMMAND TO COMPLETE
      030116  004737  043672            JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2332
 2333                                   ;GO GET REGISTER STATUS
 2334 030122  004737  043060            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      030126  000404                    BR      8$              ;GO TO 8$ IF NO ERROR
      030130  000240                    NOP                     ;RETURN HERE IF ERROR
      030132  104000                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      030134  000137  030520            JMP     21$             ;GO TO 21$ IF ERROR
 2335 030140                            8$:
 2336 030140  004737  044056            JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      030144  000405                    BR      9$              ;GO TO 9$ IF NO ERROR
      030146  000240                    NOP                     ;RETURN HERE IF ERROR
      030150  104000                    EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      030152  004736                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      030154  000137  030520            JMP     21$             ;GO TO 21$ IF ERROR
 2337 030160                            9$:
 2338 030160  004737  044710            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      030164  000405                    BR      10$             ;GO TO 10$ IF NO ERROR
      030166  000240                    NOP                     ;RETURN HERE IF ERROR
      030170  104000                    EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      030172  004736                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      030174  000137  030520            JMP     21$             ;GO TO 21$ IF ERROR
 2339 030200                            10$:
 2340
 2341                                   ;CHANGE LOOP ADDRESSES
 2342 030200  012737  030210  001124    MOV     #11$,$LPERR
 2343 030206  000410                    BR      12$
 2344
 2345                                   ;;*************************************************************************
 2346                                   ;LOOP #2        READ
 2347
 2348 030210                            11$:
 2349
 2350                                   ;PREPARE DEVICE FOR READ OPERATION
 2351 030210  004737  037316            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      030214  154130                    .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                               ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                               ;VERIFY PACK ACKNOWLEDGE
```

```
                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                        ;VERIFY RECALIBRATION
030216  000404                      BR      12$           ;GO TO 12$ IF NO ERROR
030220  000240                      NOP                   ;RETURN HERE IF ERROR
030222  104000                      EMT                   ;ERROR # DEFINED BY TSTPRP SUBROUTINE
030224  000137  030520              JMP     21$           ;GO TO 21$ IF ERROR
2352 030230                 12$:
2353
2354                        ;SET UNSAFE ERROR
2355 030230  112737  000014  001553      MOVB    #RMER1,PUTINX ;SETUP PUT INDEX TABLE
030236  112737  000200  001554          MOVB    #200,PUTINX+1 ;SET TERMINATOR BYTE
030244  012737  040000  001426          MOV     #UNS,RMER10   ;SET RMER1 OUTPUT BUFFER = UNS
030252  004737  043330                  JSR     PC,PUT        ;GO WRITE RMER1 VIA PUT SUBROUTINE
030256  000404                          BR      13$           ;GO TO 13$ IF NO ERROR
030260  000240                          NOP                   ;RETURN HERE IF ERROR
030262  104000                          EMT                   ;ERROR DEFINED BY PUT SUBROUTINE
030264  000137  030520                  JMP     21$           ;GO TO 21$ IF ERROR
2356 030270                 13$:
2357
2358                        ;READ DATA FROM DEVICE
2359 030270                 14$:
2360 030270  012737  000071  001412      MOV     #RD!GO,RMCS10         ;READ DATA COMMAND
2361 030276  012702  001553              MOV     #PUTINX,R2            ;LOAD PUT REGISTER INDEX TABLE
2362 030302  112722  000006              MOVB    #RMDA,(R2)+
2363 030306  112722  000032              MOVB    #RMOF,(R2)+
2364 030312  112722  000034              MOVB    #RMDC,(R2)+
2365 030316  112722  000004              MOVB    #RMBA,(R2)+
2366 030322  112722  000002              MOVB    #RMWC,(R2)+
2367 030326  112722  000000              MOVB    #RMCS1,(R2)+
2368 030332  112712  000200              MOVB    #200,(R2)
2369
2370 030336  004737  043330              JSR     PC,PUT        ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
030342  000404                          BR      15$           ;GO TO 15$ IF NO ERROR
030344  000240                          NOP                   ;RETURN HERE IF ERROR
030346  104000                          EMT                   ;ERROR # DEFINED BY PUT SUBROUTINE
030350  000137  030520                  JMP     21$           ;GO TO 21$ IF ERROR
2371 030354                 15$:
2372
2373                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
030354  004737  042774                  JSR     PC,GETSTS     ;GO TO GETSTS SUBROUTINE
2374
2375                        ;SEE IF DEVICE STARTED COMMAND
2376 030360  004737  043060              JSR     PC,GET        ;GO READ REGISTER(S) WITH GET SUBROUTINE
030364  000404                          BR      16$           ;GO TO 16$ IF NO ERROR
030366  000240                          NOP                   ;RETURN HERE IF ERROR
030370  104000                          EMT                   ;ERROR # DEFINED BY GET SUBROUTINE
030372  000137  030520                  JMP     21$           ;GO TO 21$ IF ERROR
2377 030376                 16$:
2378 030376  032737  020000  001350      BIT     #PIP,RMDSI            ;WAS COMMAND EXECUTED??
2379 030404  001414                      BEQ     17$                   ;NO!!
2380 030406  013737  001350  001140      MOV     RMDSI,$GDDAT          ;EXPECTED STATUS
2381 030414  042737  020000  001140      BIC     #PIP,$GDDAT
2382 030422  013737  001350  001142      MOV     RMDSI,$BDDAT          ;RECEIVED STATUS
2383 030430  104347                      EMT     347
2384 030432  000137  030520              JMP     21$
2385 030436                 17$:
2386
```

```
                                ;WAIT FOR COMMAND TO COMPLETE
030436  004737  043672                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
                                ;GO READ STATUS FOR READ OPERATION
030442  004737  043060                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
030446  000404                        BR      18$             ;GO TO 18$ IF NO ERROR
030450  000240                        NOP                     ;RETURN HERE IF ERROR
030452  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
030454  000137  030520                JMP     21$             ;GO TO 21$ IF ERROR
030460                          18$:
030460  004737  044056                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
030464  000405                        BR      19$             ;GO TO 19$ IF NO ERROR
030466  000240                        NOP                     ;RETURN HERE IF ERROR
030470  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
030472  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030474  000137  030520                JMP     21$             ;GO TO 21$ IF ERROR
030500                          19$:
030500  004737  044710                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
030504  000405                        BR      20$             ;GO TO 20$ IF NO ERROR
030506  000240                        NOP                     ;RETURN HERE IF ERROR
030510  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
030512  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030514  000137  030520                JMP     21$             ;GO TO 21$ IF ERROR
030520                          20$:

030520                          21$:

                                ;;*****************************************************************
                                ;*TEST 20        WRITE, READ EACH CURRENT LEVEL
                                ;;*****************************************************************
030520                          TST20:
030520  000004                        SCOPE                   ;SCOPE CALL
030522  000240                        NOP                     ;START OF TEST
030524  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK POINTER
030530  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
030534  013701  001466                MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
030540  012737  000020  001226        MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
030546                          1$:

                                ;;*****************************************************************
                                ;LOOP #1        FORMAT,WRITE,READ

                                ;PREPARE THE DEVICE FOR FORMAT OPERATION
030546  004737  037316                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
030552  054130                        .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                            ;VERIFY PACK ACKNOWLEDGE
                                                            ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                            ;VERIFY RECALIBRATION
030554  000404                        BR      2$              ;GO TO 2$ IF NO ERROR
030556  000240                        NOP                     ;RETURN HERE IF ERROR
030560  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
030562  000137  031556                JMP     28$             ;GO TO 28$ IF ERROR
030566                          2$:
```

```
2409                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
2410 030566 012737 000000 001446        MOV     #0,RMDCO                ;CYLINDER   0
2411 030574 012737 000000 001420 3$:    MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
2412 030602 012737 106436 001416        MOV     #BUFONE,RMBAO           ;BUS ADDRESS
2413 030610 012737 177376 001414        MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
2414 030616 012737 010000 001444        MOV     #FMT16,RMOFO            ;16 BIT FORMAT
2415 030624 012737 000063 001412        MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
2416
2417                                    ;VERIFY THAT SECTOR IS NOT BAD
     030632 004737 040242               JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     030636 000405                      BR      4$              ;GO TO 4$ IF NO ERROR
     030640 104401 070426               TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     030644 104000                      EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     030646 000137 031556               JMP     28$             ;GO TO 28$ IF ERROR
2418 030652                             4$:
2419 030652 012737 072060 001174        MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
2420 030660 012737 000001 001176        MOV     #1,$TMP1                ;RANGE OF PATTERN
2421 030666 004737 042174               JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
2422
2423                                    ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
2424 030672 012702 001553               MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
2425 030676 112722 000034               MOVB    #RMDC,(R2)+
2426 030702 112722 000006               MOVB    #RMDA,(R2)+
2427 030706 112722 000004               MOVB    #RMBA,(R2)+
2428 030712 112722 000002               MOVB    #RMWC,(R2)+
2429 030716 112722 000032               MOVB    #RMOF,(R2)+
2430 030722 112722 000000               MOVB    #RMCS1,(R2)+
2431 030726 112712 000200               MOVB    #200,(R2)               ;TERMINATE TABLE
2432 030732                             5$:
2433 030732 004737 043330               JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     030736 000404                      BR      6$              ;GO TO 6$ IF NO ERROR
     030740 000240                      NOP                     ;RETURN HERE IF ERROR
     030742 104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     030744 000137 031556               JMP     28$             ;GO TO 28$ IF ERROR
2434 030750                             6$:
2435
2436                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     030750 004737 042774               JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2437
2438                                    ;WAIT FOR COMMAND TO COMPLETE
     030754 004737 043672               JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2439
2440                                    ;GO READ STATUS FOR FORMAT OPERATION
2441 030760 004737 043060               JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     030764 000404                      BR      7$              ;GO TO 7$ IF NO ERROR
     030766 000240                      NOP                     ;RETURN HERE IF ERROR
     030770 104000                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     030772 000137 031556               JMP     28$             ;GO TO 28$ IF ERROR
2442 030776                             7$:
2443
2444                                    ;VERIFY NO ERRORS DURING FORMAT
2445 030776 004737 056572               JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     031002 000405                      BR      8$              ;GO TO 8$ IF NO ERROR
     031004 000240                      NOP                     ;RETURN HERE IF ERROR
     031006 104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     031010 004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031012 000137 031556               JMP     28$             ;GO TO 28$ IF ERROR
```

```
 2446 031016                          8$:
 2447
 2448                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2449 031016  012737  031026  001124          MOV     #9$,$LPERR
 2450 031024  000410                        BR      10$                     ;SKIP TO WRITE OPERATION
 2451
 2452                                 ;;*****************************************************************
 2453                                 ;LOOP #2        WRITE,READ
 2454
 2455 031026                          9$:
 2456
 2457
 2458                                 ;PREPARE DEVICE FOR WRITE OPERATION
 2459 031026  00473/  037316                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      031032  054130                        .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                     ;VERIFY RECALIBRATION
      031034  000404                        BR      10$             ;GO TO 10$ IF NO ERROR
      031036  000240                        NOP                     ;RETURN HERE IF ERROR
      031040  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      031042  000137  031556                JMP     28$             ;GO TO 28$ IF ERROR
 2460 031046                          10$:
 2461
 2462 031046                          11$:
 2463
 2464                                 ;WRITE DATA TO THE DRIVE
 2465 031046  012737  177400  001414          MOV     #-256.,RMWCO            ;256. WORDS (2'S COMP)
 2466 031054  012737  106442  001416          MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
 2467 031062  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2468 031070  012702  001553                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2469 031074  112722  000006                MOVB    #RMDA,(R2)+
 2470 031100  112722  000034                MOVB    #RMDC,(R2)+
 2471 031104  112722  000032                MOVB    #RMOF,(R2)+
 2472 031110  112722  000004                MOVB    #RMBA,(R2)+
 2473 031114  112722  000002                MOVB    #RMWC,(R2)+
 2474 031120  112722  000000                MOVB    #RMCS1,(R2)+
 2475 031124  112722  000200                MOVB    #200,(R2)+              ;TERMINATE TABLE
 2476
 2477 031130  004737  043330                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      031134  000404                        BR      12$             ;GO TO 12$ IF NO ERROR
      031136  000240                        NOP                     ;RETURN HERE IF ERROR
      031140  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      031142  000137  031556                JMP     28$             ;GO TO 28$ IF ERROR
 2478 031146                          12$:
 2479
 2480                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      031146  004737  042774                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2481
 2482                                 ;WAIT FOR COMMAND TO COMPLETE
      031152  004737  043672                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2483
 2484                                 ;GO READ STATUS FOR WRITE COMMAND
 2485 031156  004737  043060                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
```

```
        031162  000404                          BR      13$             ;GO TO 13$ IF NO ERROR
        031164  000240                          NOP                     ;RETURN HERE IF ERROR
        031166  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        031170  000137  031556                  JMP     28$             ;GO TO 28$ IF ERROR
 2486   031174                          13$:
 2487
 2488                                   ;CHECK FOR ERRORS DURING WRITE OPERATION
 2489   031174  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        031200  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
        031202  000240                          NOP                     ;RETURN HERE IF ERROR
        031204  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        031206  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031210  000137  031556                  JMP     28$             ;GO TO 28$ IF ERROR
 2490   031214                          14$:
 2491   031214  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        031220  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
        031222  000240                          NOP                     ;RETURN HERE IF ERROR
        031224  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        031226  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031230  000137  031556                  JMP     28$             ;GO TO 28$ IF ERROR
 2492   031234                          15$:
 2493
 2494   031234                          16$:
 2495   031234  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        031240  000405                          BR      17$             ;GO TO 17$ IF NO ERROR
        031242  000240                          NOP                     ;RETURN HERE IF ERROR
        031244  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        031246  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031250  000137  031556                  JMP     28$             ;GO TO 28$ IF ERROR
 2496   031254                          17$:
 2497
 2498                                   ;CHANGE LOOP ADDRESSES
 2499   031254  012737  031264  001124          MOV     #18$,$LPERR
 2500   031262  000410                          BR      19$                     ;SKIP TO NEXT OPERATION
 2501
 2502                                   ;;*****************************************************************
 2503                                   ;LOOP #3        READ
 2504
 2505   031264                          18$:
 2506
 2507                                   ;PREPARE DEVICE FOR READ OPERATION
 2508   031264  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        031270  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        031272  000404                          BR      19$             ;GO TO 19$ IF NO ERROR
        031274  000240                          NOP                     ;RETURN HERE IF ERROR
        031276  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        031300  000137  031556                  JMP     28$             ;GO TO 28$ IF ERROR
 2509   031304                          19$:
 2510
 2511   031304                          20$:
 2512
```

```
2513                                    ;READ DATA FROM DEVICE
2514 031304  012737  107446  001416            MOV     #BUFTWO+4,RMBAO         ;CHANGE ADDRESS
2515 031312  012737  177400  001414            MOV     #-256.,RMWCO            ;256. WORDS (2'S COMP)
2516 031320  012737  000071  001412            MOV     #RD!GO,RMCS1O           ;READ DATA COMMAND
2517 031326  012702  001553                     MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2518 031332  112722  000006                     MOVB    #RMDA,(R2)+
2519 031336  112722  000032                     MOVB    #RMOF,(R2)+
2520 031342  112722  000034                     MOVB    #RMDC,(R2)+
2521 031346  112722  000004                     MOVB    #RMBA,(R2)+
2522 031352  112722  000002                     MOVB    #RMWC,(R2)+
2523 031356  112722  000000                     MOVB    #RMCS1,(R2)+
2524 031362  112712  000200                     MOVB    #200,(R2)
2525
2526 031366  004737  043330                     JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031372  000404                             BR      21$             ;GO TO 21$ IF NO ERROR
     031374  000240                             NOP                     ;RETURN HERE IF ERROR
     031376  104000                             EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     031400  000137  031556                     JMP     28$             ;GO TO 28$ IF ERROR
2527 031404                             21$:
2528
2529                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     031404  004737  042774                     JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2530
2531                                    ;WAIT FOR COMMAND TO COMPLETE
     031410  004737  043672                     JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2532
2533                                    ;GO READ STATUS FOR READ OPERATION
2534 031414  004737  043060                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     031420  000404                             BR      22$             ;GO TO 22$ IF NO ERROR
     031422  000240                             NOP                     ;RETURN HERE IF ERROR
     031424  104000                             EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     031426  000137  031556                     JMP     28$             ;GO TO 28$ IF ERROR
2535 031432                             22$:
2536
2537                                    ;CHECK FOR ERRORS DURING READ OPERATION
2538 031432  004737  044056                     JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     031436  000405                             BR      23$             ;GO TO 23$ IF NO ERROR
     031440  000240                             NOP                     ;RETURN HERE IF ERROR
     031442  104000                             EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     031444  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031446  000137  031556                     JMP     28$             ;GO TO 28$ IF ERROR
2539 031452                             23$:
2540 031452  004737  056572                     JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     031456  000405                             BR      24$             ;GO TO 24$ IF NO ERROR
     031460  000240                             NOP                     ;RETURN HERE IF ERROR
     031462  104000                             EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     031464  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031466  000137  031556                     JMP     28$             ;GO TO 28$ IF ERROR
2541 031472                             24$:
2542
2543 031472                             25$:
2544 031472  004737  044710                     JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     031476  000405                             BR      26$             ;GO TO 26$ IF NO ERROR
     031500  000240                             NOP                     ;RETURN HERE IF ERROR
     031502  104000                             EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     031504  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031506  000137  031556                     JMP     28$             ;GO TO 28$ IF ERROR
```

```
031512                              26$:
031512  004737  042432                      JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
031516  106442                              .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
031520  107446                              .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
031522  000404                              BR      27$             ;GO TO 27$ IF NO ERROR
031524  000240                              NOP                     ;RETURN HERE IF ERROR
031526  104000                              EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
031530  000137  031556                      JMP     28$             ;GO TO 28$ IF ERROR
031534                              27$:
031534  062737  000200  001446              ADD     #128.,RMDCO             ;ADVANCE TO NEXT THRESHOLD
031542  023727  001446  001400              CMP     RMDCO,#768.             ;DONE??
031550  101002                              BHI     28$                     ;YES!!
031552  000137  030574                      JMP     3$                      ;DO NEXT CURRENT LEVEL
031556                              28$:

                                    ;;*****************************************************************
                                    ;;*TEST 21      WRITE, WRITE CHECK W/ CURRENT LEVEL SWITCHING
                                    ;;*****************************************************************
031556                              TST21:
031556  000004                              SCOPE                   ;SCOPE CALL
031560  000240                              NOP                     ;START OF TEST
031562  012706  001100                      MOV     #STACK,SP       ;INITIALIZE STACK POINTER
031566  013700  001276                      MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
031572  013701  001466                      MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
031576  012737  000021  001226              MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
031604                              1$:

                                    ;;*****************************************************************
                                    ;LOOP #1        FORMAT,WRITE,READ

031604  004737  037316                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
031610  054130                              .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
031612  000404                              BR      2$              ;GO TO 2$ IF NO ERROR
031614  000240                              NOP                     ;RETURN HERE IF ERROR
031616  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
031620  000137  032562                      JMP     26$             ;GO TO 26$ IF ERROR
031624                              2$:

                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
031624  012737  000577  001446              MOV     #383.,RMDCO             ;CYLINDER - 383.
031632  013737  001334  001420              MOV     LSTRK,RMDAO             ;SET LAST TRACK AND
031640  112737  000037  001420              MOVB    #31.,RMDAO              ;LAST SECTOR
031646  012737  106436  001416              MOV     #BUFONE,RMBAO           ;BUS ADDRESS
031654  012737  176774  001414              MOV     #-258.*2,RMWCO          ;2 SECTORS (2'S COMP)
031662  012737  010000  001444              MOV     #FMT16,RMOFO            ;16 BIT FORMAT
031670  012737  000063  001412              MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND

                                    ;VERIFY THAT SECTOR IS NOT BAD
031676  004737  040242                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
031702  000405                              BR      3$              ;GO TO 3$ IF NO ERROR
031704  104401  070426                      TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
```

```
        031710  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        031712  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
 2573   031716                          3$:
 2574   031716  012737  072016  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
 2575   031724  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
 2576   031732  004737  042174                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
 2577
 2578                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
 2579   031736  012702  001553                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
 2580   031742  112722  000034                  MOVB    #RMDC,(R2)+
 2581   031746  112722  000006                  MOVB    #RMDA,(R2)+
 2582   031752  112722  000004                  MOVB    #RMBA,(R2)+
 2583   031756  112722  000002                  MOVB    #RMWC,(R2)+
 2584   031762  112722  000032                  MOVB    #RMOF,(R2)+
 2585   031766  112722  000000                  MOVB    #RMCS1,(R2)+
 2586   031772  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
 2587   031776                          4$:
 2588   031776  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        032002  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
        032004  000240                          NOP                     ;RETURN HERE IF ERROR
        032006  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        032010  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
 2589   032014                          5$:
 2590
 2591                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        032014  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2592
 2593                                  ;WAIT FOR COMMAND TO COMPLETE
        032020  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2594
 2595                                  ;GO READ STATUS FOR FORMAT OPERATION
 2596   032024  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        032030  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        032032  000240                          NOP                     ;RETURN HERE IF ERROR
        032034  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        032036  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
 2597   032042                          6$:
 2598
 2599                                  ;VERIFY NO ERRORS DURING FORMAT
 2600   032042  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        032046  000405                          BR      7$              ;GO TO 7$ IF NO ERROR
        032050  000240                          NOP                     ;RETURN HERE IF ERROR
        032052  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        032054  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032056  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
 2601   032062                          7$:
 2602
 2603                                  ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2604   032062  012737  032072  001124          MOV     #8$,$LPERR
 2605   032070  000410                          BR      9$                      ;SKIP TO WRITE OPERATION
 2606
 2607                                  ;;*************************************************************************
 2608                                  ;LOOP #2        WRITE,READ
 2609
 2610   032072                          8$:
 2611
 2612
```

```
                                        ;PREPARE DEVICE FOR WRITE OPERATION
032072  004737  037316                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
032076  054130                                  .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
032100  000404                                  BR      9$              ;GO TO 9$ IF NO ERROR
032102  000240                                  NOP                     ;RETURN HERE IF ERROR
032104  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
032106  000137  032562                          JMP     26$             ;GO TO 26$ IF ERROR
032112                                  9$:

032112                                  10$:

                                        ;WRITE DATA TO THE DRIVE
032112  012737  177000  001414                  MOV     #-256.*2,RMWCO          ;2 SECTORS (2'S COMP)
032120  012737  106436  001416                  MOV     #BUFONE,RMBAO           ;CHANGE ADDRESS
032126  012737  000061  001412                  MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
032134  012737  072016  001174                  MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
032142  012737  000001  001176                  MOV     #1,$TMP1                ;RANGE OF PATTERN
032150  004737  042174                          JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
032154  012702  001553                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
032160  112722  000006                          MOVB    #RMDA,(R2)+
032164  112722  000034                          MOVB    #RMDC,(R2)+
032170  112722  000032                          MOVB    #RMOF,(R2)+
032174  112722  000004                          MOVB    #RMBA,(R2)+
032200  112722  000002                          MOVB    #RMWC,(R2)+
032204  112722  000000                          MOVB    #RMCS1,(R2)+
032210  112722  000200                          MOVB    #200,(R2)+              ;TERMINATE TABLE

032214  004737  043330                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
032220  000404                                  BR      11$             ;GO TO 11$ IF NO ERROR
032222  000240                                  NOP                     ;RETURN HERE IF ERROR
032224  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
032226  000137  032562                          JMP     26$             ;GO TO 26$ IF ERROR
032232                                  11$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
032232  004737  042774                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
032236  004737  043672                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE COMMAND
032242  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
032246  000404                                  BR      12$             ;GO TO 12$ IF NO ERROR
032250  000240                                  NOP                     ;RETURN HERE IF ERROR
032252  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
032254  000137  032562                          JMP     26$             ;GO TO 26$ IF ERROR
032260                                  12$:

                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
032260  004737  044056                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
032264  000405                                  BR      13$             ;GO TO 13$ IF NO ERROR
```

```
        032266  000240                          NOP                     ;RETURN HERE IF ERROR
        032270  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        032272  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032274  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
  2648  032300                          13$:
  2649  032300  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        032304  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
        032306  000240                          NOP                     ;RETURN HERE IF ERROR
        032310  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        032312  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032314  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
  2650  032320                          14$:
  2651
  2652  032320                          15$:
  2653  032320  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        032324  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        032326  000240                          NOP                     ;RETURN HERE IF ERROR
        032330  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        032332  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032334  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
  2654  032340                          16$:
  2655
  2656                                  ;CHANGE LOOP ADDRESSES
  2657  032340  012737  032350  001124          MOV     #17$,$LPERR
  2658  032346  000410                          BR      18$                             ;SKIP TO NEXT OPERATION
  2659
  2660                                  ;;*****************************************************************
  2661                                  ;LOOP #3        READ
  2662
  2663  032350                          17$:
  2664
  2665                                  ;PREPARE DEVICE FOR READ OPERATION
  2666  032350  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        032354  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        032356  000404                          BR      18$             ;GO TO 18$ IF NO ERROR
        032360  000240                          NOP                     ;RETURN HERE IF ERROR
        032362  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        032364  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
  2667  032370                          18$:
  2668
  2669  032370                          19$:
  2670
  2671                                  ;READ DATA FROM DEVICE
  2672  032370  012737  000051  001412          MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
  2673  032376  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  2674  032402  112722  000006                  MOVB    #RMDA,(R2)+
  2675  032406  112722  000032                  MOVB    #RMOF,(R2)+
  2676  032412  112722  000034                  MOVB    #RMDC,(R2)+
  2677  032416  112722  000004                  MOVB    #RMBA,(R2)+
  2678  032422  112722  000002                  MOVB    #RMWC,(R2)+
  2679  032426  112722  000000                  MOVB    #RMCS1,(R2)+
```

```
2680 032432  112712  000200                  MOVB    #200,(R2)
2681
2682 032436  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032442  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
     032444  000240                          NOP                     ;RETURN HERE IF ERROR
     032446  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032450  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
2683 032454                          20$:
2684
2685                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     032454  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2686
2687                                 ;WAIT FOR COMMAND TO COMPLETE
     032460  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2688
2689                                 ;GO READ STATUS FOR READ OPERATION
2690 032464  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032470  000404                          BR      21$             ;GO TO 21$ IF NO ERROR
     032472  000240                          NOP                     ;RETURN HERE IF ERROR
     032474  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032476  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
2691 032502                          21$:
2692
2693                                 ;CHECK FOR ERRORS DURING READ OPERATION
2694 032502  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     032506  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
     032510  000240                          NOP                     ;RETURN HERE IF ERROR
     032512  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     032514  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032516  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
2695 032522                          22$:
2696 032522  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     032526  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
     032530  000240                          NOP                     ;RETURN HERE IF ERROR
     032532  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     032534  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032536  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
2697 032542                          23$:
2698
2699 032542                          24$:
2700 032542  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     032546  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
     032550  000240                          NOP                     ;RETURN HERE IF ERROR
     032552  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     032554  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032556  000137  032562                  JMP     26$             ;GO TO 26$ IF ERROR
2701 032562                          25$:
2702
2703 032562                          26$:
2704
2705                                 ;;*****************************************************************
                                     ;*TEST 22       WRITE, READ EARLY PEAK SHIFT
                                     ;;*****************************************************************
     032562                          TST22:
     032562  000004                          SCOPE                   ;SCOPE CALL
     032564  000240                          NOP                     ;START OF TEST
     032566  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
```

```
        032572  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        032576  013701  001466          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        032602  012737  000022  001226  MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  2706
  2707                                  ;;**************************************************************
  2708                                  ;LOOP #1        FORMAT,WRITE,READ
  2709
  2710 032610                           1$:
  2711
  2712                                  ;PREPARE THE DEVICE FOR FORMAT OPERATION
  2713 032610  004737  037316          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        032614  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                              ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                              ;CLEAR CONTROLLER & SELECT DEVICE
                                                              ;VERIFY CONTROLLER CLEAR OPERATION
                                                              ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                              ;VERIFY PACK ACKNOWLEDGE
                                                              ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                              ;VERIFY RECALIBRATION
        032616  000404                  BR      2$              ;GO TO 2$ IF NO ERROR
        032620  000240                  NOP                     ;RETURN HERE IF ERROR
        032622  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        032624  000137  033570          JMP     25$             ;GO TO 25$ IF ERROR
  2714 032630                           2$:
  2715
  2716                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  2717 032630  012737  000000  001446  MOV     #0,RMDCO                ;CYLINDER = 0
  2718 032636  012737  000000  001420  MOV     #0,RMDAO                ;TRACK = 0, SECTOR - 0
  2719 032644  012737  106436  001416  MOV     #BUFONE,RMBAO           ;BUS ADDRESS
  2720 032652  012737  177376  001414  MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
  2721 032660  012737  010000  001444  MOV     #FMT16,RMOFO            ;16 BIT FORMAT
  2722 032666  012737  000063  001412  MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
  2723
  2724                                  ;VERIFY THAT SECTOR IS NOT BAD
        032674  004737  040242          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        032700  000405                  BR      3$              ;GO TO 3$ IF NO ERROR
        032702  104401  070426          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        032706  104000                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        032710  000137  033570          JMP     25$             ;GO TO 25$ IF ERROR
  2725 032714                           3$:
  2726 032714  012737  072272  001174  MOV     #EARLY,$TMP0            ;STARTING ADDRESS OF PATTERN
  2727 032722  012737  000001  001176  MOV     #1,$TMP1                ;RANGE OF PATTERN
  2728 032730  004737  042174          JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
  2729
  2730                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
  2731 032734  012702  001553          MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
  2732 032740  112722  000034          MOVB    #RMDC,(R2)+
  2733 032744  112722  000006          MOVB    #RMDA,(R2)+
  2734 032750  112722  000004          MOVB    #RMBA,(R2)+
  2735 032754  112722  000002          MOVB    #RMWC,(R2)+
  2736 032760  112722  000032          MOVB    #RMOF,(R2)+
  2737 032764  112722  000000          MOVB    #RMCS1,(R2)+
  2738 032770  112712  000200          MOVB    #200,(R2)               ;TERMINATE TABLE
  2739 032774                           4$:
  2740 032774  004737  043330          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        033000  000404                  BR      5$              ;GO TO 5$ IF NO ERROR
        033002  000240                  NOP                     ;RETURN HERE IF ERROR
```

```
        033004  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        033006  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
  2741  033012                          5$:
  2742
  2743                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        033012  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  2744
  2745                                  ;WAIT FOR COMMAND TO COMPLETE
        033016  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  2746
  2747                                  ;GO READ STATUS FOR FORMAT OPERATION
  2748  033022  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        033026  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        033030  000240                          NOP                     ;RETURN HERE IF ERROR
        033032  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        033034  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
  2749  033040                          6$:
  2750
  2751                                  ;VERIFY NO ERRORS DURING FORMAT
  2752  033040  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        033044  000405                          BR      7$              ;GO TO 7$ IF NO ERROR
        033046  000240                          NOP                     ;RETURN HERE IF ERROR
        033050  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        033052  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        033054  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
  2753  033060                          7$:
  2754
  2755                                  ;MOVE LOOP ADDRESSES TO NEXT OPERATION
  2756  033060  012737  033070  001124          MOV     #8$,$LPERR
  2757  033066  000410                          BR      9$
  2758
  2759                                  ;;*****************************************************************
  2760                                  ;LOOP #2        WRITE,READ
  2761
  2762  033070                          8$:
  2763
  2764                                  ;PREPARE DEVICE FOR WRITE OPERATION
  2765  033070  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        033074  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        033076  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
        033100  000240                          NOP                     ;RETURN HERE IF ERROR
        033102  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        033104  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
  2766  033110                          9$:
  2767
  2768                                  ;WRITE DATA TO THE DRIVE
  2769  033110                          10$:
  2770  033110  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
  2771  033116  012737  106442  001416          MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
  2772  033124  012737  000061  001412          MOV     #WD.GO,RMCS1O           ;WRITE DATA COMMAND
```

```
2773 033132  012702  001553                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2774 033136  112722  000006                MOVB    #RMDA,(R2)+
2775 033142  112722  000034                MOVB    #RMDC,(R2)+
2776 033146  112722  000032                MOVB    #RMOF,(R2)+
2777 033152  112722  000004                MOVB    #RMBA,(R2)+
2778 033156  112722  000002                MOVB    #RMWC,(R2)+
2779 033162  112722  000000                MOVB    #RMCS1,(R2)+
2780 033166  112722  000200                MOVB    #200,(R2)+              ;TERMINATE TABLE
2781
2782 033172  004737  043330                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     033176  000404                        BR      11$             ;GO TO 11$ IF NO ERROR
     033200  000240                        NOP                     ;RETURN HERE IF ERROR
     033202  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     033204  000137  033570                JMP     25$             ;GO TO 25$ IF ERROR
2783 033210                        11$:
2784
2785                               ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     033210  004737  042774                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2786
2787                               ;WAIT FOR COMMAND TO COMPLETE
     033214  004737  043672                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2788
2789                               ;GO READ STATUS FOR WRITE COMMAND
2790 033220  004737  043060                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     033224  000404                        BR      12$             ;GO TO 12$ IF NO ERROR
     033226  000240                        NOP                     ;RETURN HERE IF ERROR
     033230  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     033232  000137  033570                JMP     25$             ;GO TO 25$ IF ERROR
2791 033236                        12$:
2792
2793                               ;CHECK FOR ERRORS DURING WRITE OPERATION
2794 033236  004737  044056                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     033242  000405                        BR      13$             ;GO TO 13$ IF NO ERROR
     033244  000240                        NOP                     ;RETURN HERE IF ERROR
     033246  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     033250  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033252  000137  033570                JMP     25$             ;GO TO 25$ IF ERROR
2795 033256                        13$:
2796 033256  004737  056572                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     033262  000405                        BR      14$             ;GO TO 14$ IF NO ERROR
     033264  000240                        NOP                     ;RETURN HERE IF ERROR
     033266  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     033270  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033272  000137  033570                JMP     25$             ;GO TO 25$ IF ERROR
2797 033276                        14$:
2798 033276  004737  044710                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     033302  000405                        BR      15$             ;GO TO 15$ IF NO ERROR
     033304  000240                        NOP                     ;RETURN HERE IF ERROR
     033306  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     033310  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033312  000137  033570                JMP     25$             ;GO TO 25$ IF ERROR
2799 033316                        15$:
2800
2801                               ;CHANGE LOOP ADDRESSES
2802 033316  012737  033326  001124        MOV     #16$,$LPERR
2803 033324  000410                        BR      17$
2804
```

```
                                        ;;*****************************************************************
                                        ;LOOP #3        READ

033326                                  16$:

                                        ;PREPARE DEVICE FOR READ OPERATION
033326  004737  037316                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
033332  154130                                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
033334  000404                                  BR      17$             ;GO TO 17$ IF NO ERROR
033336  000240                                  NOP                     ;RETURN HERE IF ERROR
033340  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
033342  000137  033570                          JMP     25$             ;GO TO 25$ IF ERROR
033346                                  17$:

                                        ;READ DATA FROM DEVICE
033346                                  18$:
033346  012737  000071  001412                  MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
033354  012737  107446  001416                  MOV     #BUFTWO-4,RMBAO         ;CHANGE BUFFER
033362  012702  001553                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
033366  112722  000006                          MOVB    #RMDA,(R2)+
033372  112722  000032                          MOVB    #RMOF,(R2)+
033376  112722  000034                          MOVB    #RMDC,(R2)+
033402  112722  000004                          MOVB    #RMBA,(R2)+
033406  112722  000002                          MOVB    #RMWC,(R2)+
033412  112722  000000                          MOVB    #RMCS1,(R2)+
033416  112712  000200                          MOVB    #200,(R2)

033422  004737  043330                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
033426  000404                                  BR      19$             ;GO TO 19$ IF NO ERROR
033430  000240                                  NOP                     ;RETURN HERE IF ERROR
033432  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
033434  000137  033570                          JMP     25$             ;GO TO 25$ IF ERROR
033440                                  19$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
033440  004737  042774                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
033444  004737  043672                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR READ OPERATION
033450  004737  043060                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
033454  000404                                  BR      20$             ;GO TO 20$ IF NO ERROR
033456  000240                                  NOP                     ;RETURN HERE IF ERROR
033460  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
033462  000137  033570                          JMP     25$             ;GO TO 25$ IF ERROR
033466                                  20$:

                                        ;CHECK FOR ERRORS DURING READ OPERATION
033466  004737  044056                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
```

```
033472  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
033474  000240                          NOP                     ;RETURN HERE IF ERROR
033476  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
033500  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
033502  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
2840 033506                     21$:
2841 033506  004737  056572            JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
033512  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
033514  000240                          NOP                     ;RETURN HERE IF ERROR
033516  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
033520  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
033522  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
2842 033526                     22$:
2843 033526  004737  044710            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
033532  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
033534  000240                          NOP                     ;RETURN HERE IF ERROR
033536  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
033540  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
033542  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
2844 033546                     23$:
2845 033546  004737  042432            JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
033552  106442                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
033554  107446                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
033556  000404                          BR      24$             ;GO TO 24$ IF NO ERROR
033560  000240                          NOP                     ;RETURN HERE IF ERROR
033562  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
033564  000137  033570                  JMP     25$             ;GO TO 25$ IF ERROR
2846 033570                     24$:
2847
2848 033570                     25$:
2849
2850                            ;;*******************************************************************
                                ;*TEST 23        WRITE, READ MIXED PEAK SHIFT
                                ;;*******************************************************************
033570                          TST23:
033570  000004                          SCOPE                   ;SCOPE CALL
033572  000240                          NOP                     ;START OF TEST
033574  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
033600  013700  001276                  MOV     $BASE,R0        ;R0 - UNIBUS ADDRESS
033604  013701  001466                  MOV     TSTQUE,R1       ;(R1) - DEVICE BEING TESTED
033610  012737  000023  001226          MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2851
2852                            ;;*******************************************************************
2853                            ;LOOP #1        FORMAT,WRITE,READ
2854
2855 033616                     1$:
2856
2857                            ;PREPARE THE DEVICE FOR FORMAT OPERATION
2858 033616  004737  037316            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
033622  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
```

```
        033624  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
        033626  000240                          NOP                     ;RETURN HERE IF ERROR
        033630  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        033632  000137  034576                  JMP     25$             ;GO TO 25$ IF ERROR
  2859  033636                          2$:
  2860
  2861                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  2862  033636  012737  000000  001446          MOV     #0,RMDC0                ;CYLINDER = 0
  2863  033644  012737  000000  001420          MOV     #0,RMDA0                ;TRACK = 0, SECTOR = 0
  2864  033652  012737  106436  001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
  2865  033660  012737  177376  001414          MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
  2866  033666  012737  010000  001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
  2867  033674  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
  2868
  2869                                  ;VERIFY THAT SECTOR IS NOT BAD
        033702  004737  040242                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        033706  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        033710  104401  070426                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        033714  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        033716  000137  034576                  JMP     25$             ;GO TO 25$ IF ERROR
  2870  033722                          3$:
  2871  033722  012737  071756  001174          MOV     #MIXED,$TMP0            ;STARTING ADDRESS OF PATTERN
  2872  033730  012737  000200  001176          MOV     #128.,$TMP1             ;RANGE OF PATTERN
  2873  033736  004737  042174                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
  2874
  2875                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
  2876  033742  012702  001553                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
  2877  033746  112722  000034                  MOVB    #RMDC,(R2)+
  2878  033752  112722  000006                  MOVB    #RMDA,(R2)+
  2879  033756  112722  000004                  MOVB    #RMBA,(R2)+
  2880  033762  112722  000002                  MOVB    #RMWC,(R2)+
  2881  033766  112722  000032                  MOVB    #RMOF,(R2)+
  2882  033772  112722  000000                  MOVB    #RMCS1,(R2)+
  2883  033776  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
  2884  034002                          4$:
  2885  034002  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        034006  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
        034010  000240                          NOP                     ;RETURN HERE IF ERROR
        034012  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        034014  000137  034576                  JMP     25$             ;GO TO 25$ IF ERROR
  2886  034020                          5$:
  2887
  2888                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        034020  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  2889
  2890                                  ;WAIT FOR COMMAND TO COMPLETE
        034024  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  2891
  2892                                  ;GO READ STATUS FOR FORMAT OPERATION
  2893  034030  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        034034  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        034036  000240                          NOP                     ;RETURN HERE IF ERROR
        034040  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        034042  000137  034576                  JMP     25$             ;GO TO 25$ IF ERROR
  2894  034046                          6$:
  2895
  2896                                  ;VERIFY NO ERRORS DURING FORMAT
```

```
2897 034046  004737  056572          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     034052  000405                  BR      7$              ;GO TO 7$ IF NO ERROR
     034054  000240                  NOP                     ;RETURN HERE IF ERROR
     034056  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     034060  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     034062  000137  034576          JMP     25$             ;GO TO 25$ IF ERROR
2898 034066                      7$:
2899
2900                             ;MOVE LOOP ADDRESSES TO NEXT OPERATION
2901 034066  012737  034076  001124          MOV     #8$,$LPERR
2902 034074  000410                  BR      9$                      ;SKIP TO WRITE OPERATION
2903
2904                             ;;*************************************************************************
2905                             ;LOOP #2        WRITE,READ
2906
2907 034076                      8$:
2908
2909                             ;PREPARE DEVICE FOR WRITE OPERATION
2910 034076  004737  037316          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     034102  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                             ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                             ;CLEAR CONTROLLER & SELECT DEVICE
                                                             ;VERIFY CONTROLLER CLEAR OPERATION
                                                             ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                             ;VERIFY PACK ACKNOWLEDGE
                                                             ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                             ;VERIFY RECALIBRATION
     034104  000404                  BR      9$              ;GO TO 9$ IF NO ERROR
     034106  000240                  NOP                     ;RETURN HERE IF ERROR
     034110  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     034112  000137  034576          JMP     25$             ;GO TO 25$ IF ERROR
2911 034116                      9$:
2912
2913                             ;WRITE DATA TO THE DRIVE
2914 034116                      10$:
2915 034116  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
2916 034124  012737  106442  001416          MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
2917 034132  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2918 034140  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2919 034144  112722  000006                  MOVB    #RMDA,(R2)+
2920 034150  112722  000034                  MOVB    #RMDC,(R2)+
2921 034154  112722  000032                  MOVB    #RMOF,(R2)+
2922 034160  112722  000004                  MOVB    #RMBA,(R2)+
2923 034164  112722  000002                  MOVB    #RMWC,(R2)+
2924 034170  112722  000000                  MOVB    #RMCS1,(R2)+
2925 034174  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
2926
2927 034200  004737  043330          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     034204  000404                  BR      11$             ;GO TO 11$ IF NO ERROR
     034206  000240                  NOP                     ;RETURN HERE IF ERROR
     034210  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     034212  000137  034576          JMP     25$             ;GO TO 25$ IF ERROR
2928 034216                      11$:
2929
2930                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     034216  004737  042774          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2931
```

```
 2932                                   ;WAIT FOR COMMAND TO COMPLETE
 034222  004737  043672                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2933
 2934                                   ;GO READ STATUS FOR WRITE COMMAND
 2935 034226  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
 034232  000404                                 BR      12$             ;GO TO 12$ IF NO ERROR
 034234  000240                                 NOP                     ;RETURN HERE IF ERROR
 034236  104000                                 EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
 034240  000137  034576                         JMP     25$             ;GO TO 25$ IF ERROR
 2936 034244                            12$:
 2937
 2938                                   ;CHECK FOR ERRORS DURING WRITE OPERATION
 2939 034244  004737  044056                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
 034250  000405                                 BR      13$             ;GO TO 13$ IF NO ERROR
 034252  000240                                 NOP                     ;RETURN HERE IF ERROR
 034254  104000                                 EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
 034256  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 034260  000137  034576                         JMP     25$             ;GO TO 25$ IF ERROR
 2940 034264                            13$:
 2941 034264  004737  056572                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
 034270  000405                                 BR      14$             ;GO TO 14$ IF NO ERROR
 034272  000240                                 NOP                     ;RETURN HERE IF ERROR
 034274  104000                                 EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
 034276  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 034300  000137  034576                         JMP     25$             ;GO TO 25$ IF ERROR
 2942 034304                            14$:
 2943 034304  004737  044710                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
 034310  000405                                 BR      15$             ;GO TO 15$ IF NO ERROR
 034312  000240                                 NOP                     ;RETURN HERE IF ERROR
 034314  104000                                 EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
 034316  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 034320  000137  034576                         JMP     25$             ;GO TO 25$ IF ERROR
 2944 034324                            15$:
 2945
 2946                                   ;CHANGE LOOP ADDRESSES
 2947 034324  012737  034334  001124            MOV     #16$,$LPERR
 2948 034332  000410                            BR      17$                     ;SKIP TO NEXT OPERATION
 2949
 2950                                   ;;*****************************************************************
 2951                                   ;LOOP #3        READ
 2952
 2953 034334                            16$:
 2954
 2955                                   ;PREPARE DEVICE FOR READ OPERATION
 2956 034334  004737  037316                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
 034340  154130                                 .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
 034342  000404                                 BR      17$             ;GO TO 17$ IF NO ERROR
 034344  000240                                 NOP                     ;RETURN HERE IF ERROR
 034346  104000                                 EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
 034350  000137  034576                         JMP     25$             ;GO TO 25$ IF ERROR
```

```
2957 034354                          17$:
2958
2959                                 ;READ DATA FROM DEVICE
2960 034354                          18$:
2961 034354  012737  107446  001416        MOV     #BUFTWO+4,RMBAO          ;CHANGE BUFFER
2962 034362  012737  000071  001412        MOV     #RD!GO,RMCS10            ;READ DATA COMMAND
2963 034370  012702  001553               MOV     #PUTINX,R2               ;LOAD PUT REGISTER INDEX TABLE
2964 034374  112722  000006               MOVB    #RMDA,(R2)+
2965 034400  112722  000032               MOVB    #RMOF,(R2)+
2966 034404  112722  000034               MOVB    #RMDC,(R2)+
2967 034410  112722  000004               MOVB    #RMBA,(R2)+
2968 034414  112722  000002               MOVB    #RMWC,(R2)+
2969 034420  112722  000000               MOVB    #RMCS1,(R2)+
2970 034424  112712  000200               MOVB    #200,(R2)
2971
2972 034430  004737  043330               JSR     PC,PUT           ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     034434  000404                       BR      19$              ;GO TO 19$ IF NO ERROR
     034436  000240                       NOP                      ;RETURN HERE IF ERROR
     034440  104000                       EMT                      ;ERROR # DEFINED BY PUT SUBROUTINE
     034442  000137  034576               JMP     25$              ;GO TO 25$ IF ERROR
2973 034446                          19$:
2974
2975                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     034446  004737  042774               JSR     PC,GETSTS        ;GO TO GETSTS SUBROUTINE
2976
2977                                 ;WAIT FOR COMMAND TO COMPLETE
     034452  004737  043672               JSR     PC,TIMOUT        ;GO TO TIMOUT SUBROUTINE
2978
2979                                 ;GO READ STATUS FOR READ OPERATION
2980 034456  004737  043060               JSR     PC,GET           ;GO READ REGISTER(S) WITH GET SUBROUTINE
     034462  000404                       BR      20$              ;GO TO 20$ IF NO ERROR
     034464  000240                       NOP                      ;RETURN HERE IF ERROR
     034466  104000                       EMT                      ;ERROR # DEFINED BY GET SUBROUTINE
     034470  000137  034576               JMP     25$              ;GO TO 25$ IF ERROR
2981 034474                          20$:
2982
2983                                 ;CHECK FOR ERRORS DURING READ OPERATION
2984 034474  004737  044056               JSR     PC,PRIERR        ;GO CHECK FOR PRIMARY ERRORS
     034500  000405                       BR      21$              ;GO TO 21$ IF NO ERROR
     034502  000240                       NOP                      ;RETURN HERE IF ERROR
     034504  104000                       EMT                      ;ERROR # DEFINED BY PRIERR SUBROUTINE
     034506  004736                       JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
     034510  000137  034576               JMP     25$              ;GO TO 25$ IF ERROR
2985 034514                          21$:
2986 034514  004737  056572               JSR     PC,DTASTS        ;GO VERIFY RESULTS OF DATA TRANSFER
     034520  000405                       BR      22$              ;GO TO 22$ IF NO ERROR
     034522  000240                       NOP                      ;RETURN HERE IF ERROR
     034524  104000                       EMT                      ;ERROR # DEFINED BY DTASTS SUBROUTINE
     034526  004736                       JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
     034530  000137  034576               JMP     25$              ;GO TO 25$ IF ERROR
2987 034534                          22$:
2988 034534  004737  044710               JSR     PC,SECERR        ;GO CHECK FOR SECONDARY ERRORS
     034540  000405                       BR      23$              ;GO TO 23$ IF NO ERROR
     034542  000240                       NOP                      ;RETURN HERE IF ERROR
     034544  104000                       EMT                      ;ERROR # DEFINED BY SECERR SUBROUTINE
     034546  004736                       JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
     034550  000137  034576               JMP     25$              ;GO TO 25$ IF ERROR
```

```
 2989 034554                                23$:
 2990 034554  004737  042432                      JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
      034560  106442                              .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
      034562  107446                              .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
      034564  000404                              BR      24$             ;GO TO 24$ IF NO ERROR
      034566  000240                              NOP                     ;RETURN HERE IF ERROR
      034570  104000                              EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
      034572  000137  034576                      JMP     25$             ;GO TO 25$ IF ERROR
 2991 034576                                24$:
 2992
 2993 034576                                25$:
 2994
 2995                                       ;;*******************************************************************
                                            ;*TEST 24        WRITE, READ EACH TRACK
                                            ;;*******************************************************************
      034576                                TST24:
      034576  000004                              SCOPE                   ;SCOPE CALL
      034600  000240                              NOP                     ;START OF TEST
      034602  012706  001100                      MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      034606  013700  001276                      MOV     $BASE,R0        ;R0 - UNIBUS ADDRESS
      034612  013701  001466                      MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      034616  012737  000024  001226              MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2996
 2997                                       ;;*******************************************************************
 2998                                       ;LOOP #1        FORMAT,WRITE,READ
 2999
 3000 034624                                1$:
 3001
 3002                                       ;PREPARE THE DEVICE FOR FORMAT OPERATION
 3003 034624  004737  037316                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      034630  154130                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                           ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                           ;CLEAR CONTROLLER & SELECT DEVICE
                                                                           ;VERIFY CONTROLLER CLEAR OPERATION
                                                                           ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                           ;VERIFY PACK ACKNOWLEDGE
                                                                           ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                           ;VERIFY RECALIBRATION
      034632  000404                              BR      2$              ;GO TO 2$ IF NO ERROR
      034634  000240                              NOP                     ;RETURN HERE IF ERROR
      034636  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      034640  000137  035624                      JMP     26$             ;GO TO 26$ IF ERROR
 3004 034644                                2$:
 3005
 3006                                       ;LOAD PARAMETERS AND GENERATE DATA BUFFER
 3007 034644  012737  000000  001446              MOV     #0,RMDCO                ;CYLINDER = 0
 3008 034652  012737  000000  001420              MOV     #0,RMDAO                ;TRACK = 0, SECTOR   0
 3009 034660                                3$:
 3010 034660  012737  106436  001416              MOV     #BUFONE,RMBAO           ;BUS ADDRESS
 3011 034666  012737  177376  001414              MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
 3012 034674  012737  010000  001444              MOV     #FMT16,RMOFO            ;16 BIT FORMAT
 3013 034702  012737  000063  001412              MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
 3014
 3015                                       ;VERIFY THAT SECTOR IS NOT BAD
      034710  004737  040242                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
      034714  000405                              BR      4$              ;GO TO 4$ IF NO ERROR
      034716  104401  070426                      TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
```

```
        034722  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        034724  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
3016    034730                          4$:
3017    034730  012737  071756  001174          MOV     #MIXED,$TMP0            ;STARTING ADDRESS OF PATTERN
3018    034736  012737  000200  001176          MOV     #128.,$TMP1             ;RANGE OF PATTERN
3019    034744  004737  042774                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
3020
3021                                    ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
3022    034750  012702  001553                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
3023    034754  112722  000034                  MOVB    #RMDC,(R2)+
3024    034760  112722  000006                  MOVB    #RMDA,(R2)+
3025    034764  112722  000004                  MOVB    #RMBA,(R2)+
3026    034770  112722  000002                  MOVB    #RMWC,(R2)+
3027    034774  112722  000032                  MOVB    #RMOF,(R2)+
3028    035000  112722  000000                  MOVB    #RMCS1,(R2)+
3029    035004  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
3030    035010                          5$:
3031
3032                                    ;FORMAT THE DRIVE
3033    035010  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        035014  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        035016  000240                          NOP                     ;RETURN HERE IF ERROR
        035020  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        035022  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
3034    035026                          6$:
3035
3036                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        035026  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
3037
3038                                    ;WAIT FOR COMMAND TO COMPLETE
        035032  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
3039
3040                                    ;GO READ STATUS FOR FORMAT OPERATION
3041    035036  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        035042  000404                          BR      7$              ;GO TO 7$ IF NO ERROR
        035044  000240                          NOP                     ;RETURN HERE IF ERROR
        035046  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        035050  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
3042    035054                          7$:
3043
3044                                    ;VERIFY NO ERRORS DURING FORMAT
3045    035054  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        035060  000405                          BR      8$              ;GO TO 8$ IF NO ERROR
        035062  000240                          NOP                     ;RETURN HERE IF ERROR
        035064  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        035066  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035070  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
3046    035074                          8$:
3047
3048                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
3049    035074  012737  035104  001124          MOV     #9$,$LPERR
3050    035102  000410                          BR      10$                     ;SKIP TO WRITE OPERATION
3051
3052                                    ;;*****************************************************************
3053                                    ;LOOP #2        WRITE,READ
3054
3055    035104                          9$:
```

```
3056
3057                                    ;PREPARE DEVICE FOR WRITE OPERATION
3058 035104  004737  037316                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     035110  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
     035112  000404                            BR      10$             ;GO TO 10$ IF NO ERROR
     035114  000240                            NOP                     ;RETURN HERE IF ERROR
     035116  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     035120  000137  035624                    JMP     26$             ;GO TO 26$ IF ERROR
3059 035124                             10$:
3060
3061                                    ;WRITE DATA TO THE DRIVE
3062 035124                             11$:
3063 035124  012737  106442  001416            MOV     #BUFONE+4,RMBAO         ;CHANGE BUS ADDRESS
3064 035132  012737  177400  001414            MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
3065 035140  012737  000061  001412            MOV     #WD.GO,RMCS10           ;WRITE DATA COMMAND
3066 035146  012702  001553                    MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
3067 035152  112722  000006                    MOVB    #RMDA,(R2)+
3068 035156  112722  000034                    MOVB    #RMDC,(R2)+
3069 035162  112722  000032                    MOVB    #RMOF,(R2)+
3070 035166  112722  000004                    MOVB    #RMBA,(R2)+
3071 035172  112722  000002                    MOVB    #RMWC,(R2)+
3072 035176  112722  000000                    MOVB    #RMCS1,(R2)+
3073 035202  112722  000200                    MOVB    #200,(R2)+              ;TERMINATE TABLE
3074
3075 035206  004737  043330                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     035212  000404                            BR      12$             ;GO TO 12$ IF NO ERROR
     035214  000240                            NOP                     ;RETURN HERE IF ERROR
     035216  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     035220  000137  035624                    JMP     26$             ;GO TO 26$ IF ERROR
3076 035224                             12$:
3077
3078                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     035224  004737  042774                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
3079
3080                                    ;WAIT FOR COMMAND TO COMPLETE
     035230  004737  043672                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
3081
3082                                    ;GO READ STATUS FOR WRITE COMMAND
3083 035234  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     035240  000404                            BR      13$             ;GO TO 13$ IF NO ERROR
     035242  000240                            NOP                     ;RETURN HERE IF ERROR
     035244  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     035246  000137  035624                    JMP     26$             ;GO TO 26$ IF ERROR
3084 035252                             13$:
3085
3086                                    ;CHECK FOR ERRORS DURING WRITE OPERATION
3087 035252  004737  044056                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     035256  000405                            BR      14$             ;GO TO 14$ IF NO ERROR
     035260  000240                            NOP                     ;RETURN HERE IF ERROR
     035262  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
```

```
        035264  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035266  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3088   035272                          14$:
 3089   035272  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        035276  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
        035300  000240                          NOP                     ;RETURN HERE IF ERROR
        035302  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        035304  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035306  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3090   035312                          15$:
 3091   035312  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        035316  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        035320  000240                          NOP                     ;RETURN HERE IF ERROR
        035322  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        035324  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035326  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3092   035332                          16$:
 3093
 3094                                    ;CHANGE LOOP ADDRESSES
 3095   035332  012737  035342  001124          MOV     #17$,$LPERR
 3096   035340  000410                          BR      18$                     ;SKIP TO NEXT OPERATION
 3097
 3098                                    ;;*****************************************************************
 3099                                    ;LOOP #3        READ
 3100
 3101   035342                          17$:
 3102
 3103                                    ;PREPARE DEVICE FOR READ OPERATION
 3104   035342  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        035346  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        035350  000404                          BR      18$             ;GO TO 18$ IF NO ERROR
        035352  000240                          NOP                     ;RETURN HERE IF ERROR
        035354  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        035356  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3105   035362                          18$:
 3106
 3107                                    ;READ DATA FROM DEVICE
 3108   035362                          19$:
 3109   035362  012737  107446  001416          MOV     #BUFTWO+4,RMBAO         ;CHANGE BUS ADDRESS
 3110   035370  012737  000071  001412          MOV     #RD.GO,RMCS1O           ;READ DATA COMMAND
 3111   035376  012702  001553                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 3112   035402  112722  000006                  MOVB    #RMDA,(R2)+
 3113   035406  112722  000032                  MOVB    #RMOF,(R2)+
 3114   035412  112722  000034                  MOVB    #RMDC,(R2)+
 3115   035416  112722  000004                  MOVB    #RMBA,(R2)+
 3116   035422  112722  000002                  MOVB    #RMWC,(R2)+
 3117   035426  112722  000000                  MOVB    #RMCS1,(R2)+
 3118   035432  112712  000200                  MOVB    #200,(R2)
 3119
 3120   035436  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
```

```
        035442  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        035444  000240                          NOP                     ;RETURN HERE IF ERROR
        035446  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        035450  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3121   035454                          20$:
 3122
 3123                                   ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        035454  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 3124
 3125                                   ;WAIT FOR COMMAND TO COMPLETE
        035460  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 3126
 3127                                   ;GO READ STATUS FOR READ OPERATION
 3128   035464  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        035470  000404                          BR      21$             ;GO TO 21$ IF NO ERROR
        035472  000240                          NOP                     ;RETURN HERE IF ERROR
        035474  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        035476  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3129   035502                          21$:
 3130
 3131                                   ;CHECK FOR ERRORS DURING READ OPERATION
 3132   035502  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        035506  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
        035510  000240                          NOP                     ;RETURN HERE IF ERROR
        035512  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        035514  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035516  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3133   035522                          22$:
 3134   035522  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        035526  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
        035530  000240                          NOP                     ;RETURN HERE IF ERROR
        035532  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        035534  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035536  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3135   035542                          23$:
 3136   035542  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        035546  000405                          BR      24$             ;GO TO 24$ IF NO ERROR
        035550  000240                          NOP                     ;RETURN HERE IF ERROR
        035552  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        035554  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035556  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3137   035562                          24$:
 3138   035562  004737  042432                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        035566  106442                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        035570  107446                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        035572  000404                          BR      25$             ;GO TO 25$ IF NO ERROR
        035574  000240                          NOP                     ;RETURN HERE IF ERROR
        035576  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        035600  000137  035624                  JMP     26$             ;GO TO 26$ IF ERROR
 3139   035604                          25$:
 3140   035604  105237  001421                  INCB    RMDA0+1                 ;ADVANCE TO NEXT TRACK
 3141   035610  123737  001421  001335          CMPB    RMDA0+1,LSTRK+1         ;DONE ?
 3142   035616  101002                          BHI     26$                     ;YES!!
 3143   035620  000137  034660                  JMP     3$                      ;TEST NEXT TRACK
 3144   035624                          26$:
 3145
 3146                                   ;;*****************************************************************
```

```
                                        ;*TEST 25       READ, WRITE CHECK MULTIPLE SECTORS IN OFFSET MODE
                                        ;;*****************************************************************
      035624                            TST25:
      035624  000004                            SCOPE                   ;SCOPE CALL
      035626  000240                            NOP                     ;START OF TEST
      035630  012706  001100                    MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      035634  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      035640  013701  001466                    MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      035644  012737  000025  001226            MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 3147
 3148                                   ;;*****************************************************************
 3149                                   ;LOOP #1        FORMAT,READ OFFSET,WRITE CHECK OFFSET IN BOTH DIRECTIONS
 3150 035652                            1$:
 3151
 3152                                   ;PREPARE DEVICE FOR FORMAT OPERATION
 3153 035652  004737  037316                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      035656  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
      035660  000404                            BR      2$              ;GO TO 2$ IF NO ERROR
      035662  000240                            NOP                     ;RETURN HERE IF ERROR
      035664  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      035666  000137  037006                    JMP     28$             ;GO TO 28$ IF ERROR
 3154
 3155                                   ;LOAD PARAMETERS AND GENERATE DATA BUFFER
 3156 035672                            2$:
 3157 035672  012737  000000  001446            MOV     #0,RMDC0                ;CYLINDER = 0
 3158 035700  012737  000000  001420            MOV     #0,RMDA0                ;TRACK = 0, SECTOR   0
 3159 035706  012737  106436  001416            MOV     #BUFONE,RMBA0           ;BUFFER ADDRESS
 3160 035714  012737  176774  001414            MOV     #-258.*2,RMWC0          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
 3161 035722  012737  010000  001444            MOV     #FMT16,RMOF0            ;16 BIT MODE
 3162 035730  012737  000063  001412            MOV     #WH'GO,RMCS10           ;WRITE HEADER
 3163
 3164                                   ;VERIFY THAT SECTOR IS NOT BAD
      035736  004737  040242                    JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
      035742  000405                            BR      3$              ;GO TO 3$ IF NO ERROR
      035744  104401  070426                    TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
      035750  104000                            EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
      035752  000137  037006                    JMP     28$             ;GO TO 28$ IF ERROR
 3165 035756                            3$:
 3166 035756  012737  072060  001174            MOV     #ZEROS,$TMP0            ;DATA PATTERN
 3167 035764  012737  000001  001176            MOV     #1,$TMP1
 3168 035772  004737  042174                    JSR     PC,GENBUF               ;TO GENERATE BUFFER FOR FORMAT
 3169
 3170                                   ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
 3171 035776  012702  001553                    MOV     #PUTINX,R2              ;BYTE ENTRY TABLE
 3172 036002  112722  000034                    MOVB    #RMDC,(R2)+
 3173 036006  112722  000006                    MOVB    #RMDA,(R2)+
 3174 036012  112722  000004                    MOVB    #RMBA,(R2)+
 3175 036016  112722  000002                    MOVB    #RMWC,(R2)+
 3176 036022  112722  000032                    MOVB    #RMOF,(R2)+
 3177 036026  112722  000000                    MOVB    #RMCS1,(R2)+
```

```
                                        MOVB    #200,(R2)               ;TABLE TERMINATOR
                                4$:
                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      5$              ;GO TO 5$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     28$             ;GO TO 28$ IF ERROR
                                5$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
                                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                ;WAIT FOR COMMAND TO COMPLETE
                                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                ;GO READ STATUS FOR FORMAT OPERATION
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      6$              ;GO TO 6$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     28$             ;GO TO 28$ IF ERROR
                                6$:

                                ;VERIFY NO ERROR DURING FORMAT
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      7$              ;GO TO 7$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     28$             ;GO TO 28$ IF ERROR

                                ;MOV LOOP ADDRESSES TO NEXT OPERATION
                                7$:
                                        MOV     #8$,$LPERR      ;ERROR LOOP ADDRESS

                                ;LOCATE BUFFER ADDRESS AND WORD COUNT
                                        MOV     #-256.*2,RMWCO  ;TWO SECTORS
                                        MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS
                                ;;*****************************************************************
                                ;LOOP 2  READ AND WRITE CHECK IN OFFSET MODE
                                8$:
                                ;PREPARE DEVICE FOR READ OFFSET OPEATION
                                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
                                        BR      9$              ;GO TO 9$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                                        JMP     28$             ;GO TO 28$ IF ERROR
                                9$:
                                10$:
```

```
3208                                    ;READ OFFSET
3209 036164  012702  001553                    MOV     #PUTINX,R2      ;LOAD THE TABLE
3210 036170  112722  000006                    MOVB    #RMDA,(R2)+
3211 036174  112722  000034                    MOVB    #RMDC,(R2)+
3212 036200  112722  000032                    MOVB    #RMOF,(R2)+
3213 036204  112722  000004                    MOVB    #RMBA,(R2)+
3214 036210  112722  000002                    MOVB    #RMWC,(R2)+
3215 036214  112722  000200                    MOVB    #200,(R2)+      ;TABLE TERMINATOR
3216
3217 036220  004737  043330                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     036224  000404                            BR      11$             ;GO TO 11$ IF NO ERROR
     036226  000240                            NOP                     ;RETURN HERE IF ERROR
     036230  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     036232  000137  037006                    JMP     28$             ;GO TO 28$ IF ERROR
3218 036236  012737  000015  001412  11$:      MOV     #OFFSET.GO,RMCS10       ;OFFSET COMMAND
3219 036244  012702  001553                    MOV     #PUTINX,R2
3220 036250  112722  000000                    MOVB    #RMCS1,(R2)+
3221 036254  112722  000200                    MOVB    #200,(R2)+
3222
3223 036260  004737  043330                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     036264  000404                            BR      12$             ;GO TO 12$ IF NO ERROR
     036266  000240                            NOP                     ;RETURN HERE IF ERROR
     036270  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     036272  000137  037006                    JMP     28$             ;GO TO 28$ IF ERROR
3224 036276                             12$:
3225
3226                                    ;WAIT FOR COMMAND TO COMPLETE
     036276  004737  043672                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
3227
3228 036302  012737  000071  001412            MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
3229 036310  012702  001553                    MOV     #PUTINX,R2              ;TABLE INDEX
3230 036314  112722  000000                    MOVB    #RMCS1,(R2)+
3231 036320  112712  000200                    MOVB    #200,(R2)               ;TABLE TERMINATOR
3232
3233 036324  004737  043330                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     036330  000404                            BR      13$             ;GO TO 13$ IF NO ERROR
     036332  000240                            NOP                     ;RETURN HERE IF ERROR
     036334  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     036336  000137  037006                    JMP     28$             ;GO TO 28$ IF ERROR
3234 036342                             13$:
3235
3236                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     036342  004737  042774                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
3237
3238                                    ;WAIT FOR COMMAND TO COMPLETE
     036346  004737  043672                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
3239
3240                                    ;READ STATUS FOR READ OFFSET OPERATION
3241 036352  004737  043060                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     036356  000404                            BR      14$             ;GO TO 14$ IF NO ERROR
     036360  000240                            NOP                     ;RETURN HERE IF ERROR
     036362  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     036364  000137  037006                    JMP     28$             ;GO TO 28$ IF ERROR
3242
3243                                    ;CHECK ERROR DURING READ OFFSET
3244 036370                             14$:
3245 036370  004737  044056                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
```

```
        036374  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
        036376  000240                          NOP                     ;RETURN HERE IF ERROR
        036400  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        036402  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036404  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
3246    036410                          15$:
3247    036410  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        036414  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        036416  000240                          NOP                     ;RETURN HERE IF ERROR
        036420  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        036422  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036424  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
3248    036430                          16$:
3249    036430  004737  044710                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        036434  000405                          BR      17$             ;GO TO 17$ IF NO ERROR
        036436  000240                          NOP                     ;RETURN HERE IF ERROR
        036440  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        036442  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036444  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
3250    036450                          17$:
3251
3252                                    ;CHANGE LOOP ADDRESS
3253    036450  012737  036460  001124          MOV     #18$,$LPERR
3254    036456  000410                          BR      19$             ;TO NEXT OPERATION
3255
3256                                    ;;*****************************************************************
3257                                    ;LOOP 3 WRITE CHECK OFFSET
3258    036460                          18$:
3259
3260                                    ;PREPARE DEVICE FOR READ OPERATION
3261    036460  004737  037316                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        036464  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        036466  000404                          BR      19$             ;GO TO 19$ IF NO ERROR
        036470  000240                          NOP                     ;RETURN HERE IF ERROR
        036472  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        036474  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
3262    036500                          19$:
3263
3264                                    ;WRITE CHECK DATA IN OFFSET MODE
3265    036500                          20$:
3266    036500  012702  001553                  MOV     #PUTINX,R2
3267    036504  112722  000006                  MOVB    #RMDA,(R2)+
3268    036510  112722  000032                  MOVB    #RMOF,(R2)+
3269    036514  112722  000034                  MOVB    #RMDC,(R2)+
3270    036520  112722  000004                  MOVB    #RMBA,(R2)+
3271    036524  112722  000002                  MOVB    #RMWC,(R2)+
3272    036530  112712  000200                  MOVB    #200,(R2)       ;TABLE TERMINATOR
3273
3274    036534  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        036540  000404                          BR      21$             ;GO TO 21$ IF NO ERROR
```

```
        036542  000240                          NOP                     ;RETURN HERE IF ERROR
        036544  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        036546  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3275
  3276                                  ;SETUP OFFSET MODE FIRST
  3277 036552                           21$:
  3278 036552  012737  000015  001412          MOV     #OFFSET!GO,RMCS10       ;OUTPUT BUFFER
  3279 036560  012702  001553                  MOV     #PUTINX,R2
  3280 036564  112722  000000                  MOVB    #RMCS1,(R2)+
  3281 036570  112712  000200                  MOVB    #200,(R2)
  3282
  3283 036574  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        036600  000404                          BR      22$             ;GO TO 22$ IF NO ERROR
        036602  000240                          NOP                     ;RETURN HERE IF ERROR
        036604  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        036606  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3284 036612                           22$:
  3285
  3286                                  ;WAIT FOR COMMAND TO COMPLETE
        036612  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  3287
  3288                                  ;EXECUTE WRITE CHECK DATA COMMAND
  3289 036616  012737  000051  001412          MOV     #WCD!GO,RMCS10
  3290 036624  012702  001553                  MOV     #PUTINX,R2
  3291 036630  112722  000000                  MOVB    #RMCS1,(R2)+
  3292 036634  112712  000200                  MOVB    #200,(R2)
  3293
  3294 036640  004737  043330                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        036644  000404                          BR      23$             ;GO TO 23$ IF NO ERROR
        036646  000240                          NOP                     ;RETURN HERE IF ERROR
        036650  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        036652  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3295 036656                           23$:
  3296
  3297                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        036656  004737  042774                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  3298
  3299                                  ;WAIT FOR COMMAND TO COMPLETE
        036662  004737  043672                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  3300
  3301                                  ;READ STATUS FOR WRITE CHECK OFFSET OPERATION
  3302 036666  004737  043060                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        036672  000404                          BR      24$             ;GO TO 24$ IF NO ERROR
        036674  000240                          NOP                     ;RETURN HERE IF ERROR
        036676  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        036700  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3303 036704                           24$:
  3304
  3305                                  ;CHECK ERROR DURING WRITE CHECK OFFSET OPERATION
  3306 036704  004737  044056                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        036710  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
        036712  000240                          NOP                     ;RETURN HERE IF ERROR
        036714  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        036716  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036720  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3307 036724                           25$:
  3308 036724  004737  056572                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
```

```
        036730  000405                          BR      26$             ;GO TO 26$ IF NO ERROR
        036732  000240                          NOP                     ;RETURN HERE IF ERROR
        036734  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        036736  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036740  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3309  036744                          26$:
  3310  036744  004737  044710                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        036750  000405                          BR      27$             ;GO TO 27$ IF NO ERROR
        036752  000240                          NOP                     ;RETURN HERE IF ERROR
        036754  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        036756  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036760  000137  037006                  JMP     28$             ;GO TO 28$ IF ERROR
  3311  036764                          27$:
  3312  036764  032737  000200  001444          BIT     #OFD,RMOFO      ;IN FORWARD DIRECTION OF OFFSET ?
  3313  036772  001005                          BNE     28$             ;BRANCH IF SO
  3314  036774  052737  000200  001444          BIS     #OFD,RMOFO      ;SET TO FORWARD DIRECTION
  3315  037002  000137  036122                  JMP     7$              ;CHANGE OFFSET DIRECTION TEST AGAIN
  3316  037006                          28$:
```

```
                                        .SBTTL  END OF SUB-PASS ROUTINE

                                        ;THIS IS THE END OF SUB-PASS ROUTINE. THIS ROUTINE IS USED TO
                                        ;TERMINATE THE OPERATION OF THE CURRENT DEVICE UNDER TEST AND
                                        ;SELECT THE NEXT DEVICE FOR TEST. IF THERE ARE NO MORE DEVICES
                                        ;TO TEST, EXIT IS MADE TO '$EOP' ROUTINE. OTHERWISE, RETURN
                                        ;IS MADE TO 'SHUT' ROUTINE.

037006  000004                  $EOSP:  SCOPE
037010  000240                          NOP
037012  013700  001466                  MOV     TSTQUE,R0       ;GET POINTER TO TSTQUE
037016  062700  000002                  ADD     #2,R0           ;ADJUST POINTER TO NEXT DEVICE
037022  010037  001466                  MOV     R0,TSTQUE       ;SAVE POINTER TO TSTQUE
037026  005710                          TST     (R0)            ;ANY MORE DEVICES FOR TEST ?
037030  001402                          BEQ     1$              ;BR IF NO
037032  000137  037262                  JMP     SHUT            ;JUMP TO 'SHUT' ROUTINE
037036  012737  001470  001466  1$:     MOV     #TSTQUE+2,TSTQUE        ;INITIALIZE POINTER TO FIRST DEVICE IN
                                                                ;TEST QUE TABLE

                                        .SBTTL  END OF PASS ROUTINE

                                        ;;**********************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY''
                                        ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO SHUT

037044                          $EOP:
037044  000240                          NOP
037046  005037  001116                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
037052  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
037056  005237  001230                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
037062  042737  100000  001230          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
037070  005327                          DEC     (PC)+           ;;LOOP?
037072  000001                  $EOPCT: .WORD   1
037074  003066                          BGT     $DOAGN          ;;YES
037076  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
037100  000001                  $ENDCT: .WORD   1
037102  037072                          $EOPCT
037104  104401  037112                  TYPE    ,65$            ;;TYPE ASCIZ STRING
037110  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
037130                          64$:
037130  013746  001230                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
037134  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037136  005737  001126                  TST     $ERTTL          ;;SEE IF ANY ERRORS THIS PASS
037142  001431                          BEQ     $GT42P          ;;BR IF NO ERRORS TO REPORT
037144  104401  037152                  TYPE    ,67$            ;;TYPE ASCIZ STRING
037150  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
037214                          66$:
037214  013746  001126                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
037220  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037222  005037  001126                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
```

```
037226  104401  001217          $GT42P: TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
037232  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
037236  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
037240  000005                          RESET                   ;;CLEAR THE WORLD
037242  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
037244  000240                          NOP                     ;;SAVE ROOM
037246  000240                          NOP                     ;;FOR
037250  000240                          NOP                     ;;ACT11
037252                          $DOAGN:
037252  000137                          JMP     @(PC)+          ;;RETURN
037254  037262                  $RTNAD: .WORD   SHUT
037256     377      377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        .EVEN

    21
    22 037262  005737  001326          SHUT:   TST     CTLFG           ;WAS CONTROL C FLAGGED ?
    23 037266  001002                          BNE     1$              ;BR IF YES
    24 037270  000137  007670                  JMP     READY           ;CONTINUE
    25 037274  005737  000042          1$:     TST     @#42            ;ANY MONITOR PRESENT ?
    26 037300  001002                          BNE     2$              ;BR IF YES
    27 037302  000137  005432                  JMP     START           ;GO TO START
    28 037306  005037  001300          2$:     CLR     $DEVM           ;FUDGE NO DRIVES IN MAP
    29 037312  000137  037044                  JMP     $EOP            ;RETURN TO $EOP
```

```
                        .SBTTL  TEST PREPARATION MODULE

                        ; THIS MODULE PREPARES THE SUBSYSTEM FOR THE EXECUTION OF A TEST,
                        ; REPORTING AN ERROR TO THE USER IF AN ERROR IS  DETECTED.  THE USER
                        ; SPECIFIES TASKS TO BE PERFORMED, WHICH THE MODULE EXECUTES
                        ; USING SUBROUTINES.

                        ;CALL:
                        ;       JSR     PC,TSTPRP
                        ;       .WORD   NN'NNN          TASK/VERIFY DESCRIPTOR
                        ;       BR      ??              RETURN HERE IF NO ERROR
                        ;       NOP                     RETURN HERE IF ERROR
                        ;       ERROR                   ERROR DEFINED BY MODULE

                        ;TASK/VERIFY DESCRIPTOR
                        ;       BIT 15 = 1      SELECT DEVICE AND VERIFY DEVICE IS AVAILABLE
                        ;       ----------
                        ;       BIT 14 = 1      CLEAR CONTROLLER AND SELECT DEVICE
                        ;       BIT 13          (RESERVED FOR DRIVE CLEAR)
                        ;       BIT 12 - 1      PACK ACKNOWLEDGE IF VOLUME NOT VALID
                        ;       ----------
                        ;       BIT 11 - 1      RECALIBRATE IF POSITIONING IN PROGRESS
                        ;       BIT 10
                        ;       BIT 9
                        ;       ----------
                        ;       BIT 8
                        ;       BIT 7
                        ;       BIT 6  = 1      VERIFY CONTROLLER CLEAR OPERATION
                        ;       ----------
                        ;       BIT 5           (RESERVED FOR DRIVE CLEAR)
                        ;       BIT 4  = 1      VERIFY PACK ACKNOWLEDGE
                        ;       BIT 3  = 1      VERIFY RECALIBRATION
                        ;       ----------
                        ;       BIT 2
                        ;       BIT 1
                        ;       BIT 0

037316                  TSTPRP:

                        ;STORE TASK DESCRIPTOR AND CLEAR USER'S ERROR CALL
037316  017637  000000  040236          MOV     @(SP),500$      ;STORE DESCRIPTOR
037324  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
037330  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR CALL
037334  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN

037340  004737  042774                  JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
037344  004737  043060                  JSR     PC,GET          ;GET RMER2
037350  000411                          BR      15$             ;BR IF NO ERROR DETECTED
037352  000401                          BR      10$             ;GET OVER ERROR NUMBER
037354  000000                          .WORD   0               ;ERROR DEFINED BY GET SUBROUTINE
037356  062716  000004          10$:    ADD     #4,(SP)         ;XFER ERROR TO USER AND
037362  113776  037354  000000          MOVB    10$-2,@(SP)     ;GET ERROR NUMBER.
037370  000137  040226                  JMP     400$

037374  013737  001400  040240  15$:    MOV     RMER2I,505$     ;GET RMER2 AND SAVE FOR LATER

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

```
                                        ;SELECT DEVICE AND VERIFY DEVICE AVAILABLE IF BIT 15 SET IN TASK
037402  005737  040236                        TST     500$            ;SELECT DEVICE??
037406  100014                                BPL     30$             ;NO!!

037410  004737  050762                        JSR     PC,DEVSEL       ;GO SELECT DEVICE
037414  000411                                BR      30$             ;NO ERROR - CONTINUE
037416  000401                                BR      20$
037420  000000                                .WORD   0               ;ERROR NUMBER FROM DEVSEL
037422  062716  000004              20$:      ADD     #4,(SP)         ;TRANSFER ERROR TO USER
037426  113776  037420  000000              MOVB    20$-2,@(SP)
037434  000137  040226                        JMP     400$

                                        ;;*****************************************************************
                                        ;CLEAR CONTROLLER IF BIT 14 IS SET IN TASK
037440                                  30$:
037440  032737  040000  040236              BIT     #BIT14,500$     ;CLEAR CONTROLLER??
037446  001451                                BEQ     120$            ;NO!!

037450  004737  052434                        JSR     PC,CNTCLR       ;GO CLEAR CONTROLLER
037454  000411                                BR      60$             ;CONTINUE - NO ERROR
037456  000401                                BR      50$
037460  000000                          40$:  .WORD   0               ;ERROR NUMBER FROM CNTCLR
037462  062716  000004              50$:      ADD     #4,(SP)         ;TRANSFER ERROR TO USER
037466  113776  037460  000000              MOVB    40$,@(SP)
037474  000137  040226                        JMP     400$

                                        ;;*****************************************************************
                                        ;VERIFY CONTROLLER CLEAR IF BIT6 SET IN TASK
037500                                  60$:
037500  032737  000100  040236              BIT     #BIT6,500$      ;VERIFY??
037506  001431                                BEQ     120$            ;NO!!

037510  004737  043060                        JSR     PC,GET          ;GO GET STATUS
037514  000411                                BR      90$             ;NO ERROR GETTING STATUS
037516  000401                                BR      80$
037520  000000                          70$:  .WORD   0               ;ERROR FROM GETTING STATUS
037522  062716  000004              80$:      ADD     #4,(SP)         ;TRANSFER ERROR TO USER
037526  113776  037520  000000              MOVB    70$,@(SP)
037534  000137  040226                        JMP     400$

037540  004737  052552              90$:      JSR     PC,CLRSTS       ;GO VERIFY STATUS CLEAR
037544  000412                                BR      120$            ;NO FRROR IN CLEAR
037546  000401                                BR      110$
037550  000000                         100$:  .WORD   0               ;ERROR IN STATUS CLEAR
037552  005726                         110$:  TST     (SP)+           ;STRIP RETURN ADDRESS TO
037554  062716  000004                        ADD     #4,(SP)         ;SUBROUTINE AND TRANSFER
037560  113776  037550  000000              MOVB    100$,@(SP)      ;ERROR TO USER
037566  000137  040226                        JMP     400$

                                        ;;*****************************************************************
                                        ;EXECUTE PACK ACKNOWLEDGE IF BIT12 SET IN TASK AND VOLUME IS
                                        ;NOT VALID
037572                                  120$:
037572  032737  010000  040236              BIT     #BIT12,500$     ;PACK ACKNOWLEDGE??
037600  001503                                BEQ     240$            ;NO.!

037602  004737  043060                        JSR     PC,GET
```

```
                                                BR      150$            ;NO ERROR GETTING RMDS
                                                BR      140$
                                        130$:   .WORD   0
                                        140$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                                                MOVB    130$,@(SP)
                                                JMP     400$

                                        150$:   BIT     #VV,RMDSI       ;IS VOLUME VALID??
                                                BNE     240$            ;YES!.

                                                CLR     MEDENB          ;CLEAR MEDIA ENABLE
                                                MOV     #PAKACK!GO,RMCS10       ;LOAD PACK ACK COMMAND
                                                MOVB    #RMCS1,PUTINX   ;SETUP REGISTER INDEX TABLE
                                                MOVB    #200,PUTINX+1
                                                JSR     PC,PUT          ;GO WRITE COMMAND
                                                BR      180$            ;NO ERROR LOADING REGISTER
                                                BR      170$
                                        160$:   .WORD   0               ;ERROR FROM PUT SUB
                                        170$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                                                MOVB    160$,@(SP)
                                                BR      400$

                                        180$:   JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

                                        ;;**************************************************************************
                                        ;VERIFY PACK ACKNOWLEDGE IF #BIT4 SET IN TASK
                                                BIT     #BIT4,500$      ;VERIFY PACK ACKNOWLEDGE??
                                                BEQ     240$            ;NO!.

                                                JSR     PC,GET          ;GO GET STATUS
                                                BR      210$            ;NO ERROR GETTING STATUS
                                                BR      200$
                                        190$:   .WORD   0               ;ERROR FROM GET SUB
                                        200$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                                                MOVB    190$,@(SP)
                                                BR      400$

                                        210$:   JSR     PC,ACKSTS       ;GO CHECK ACKNOWLEDGE
                                                BR      240$            ;NO ERROR
                                                BR      230$
                                        220$:   .WORD   0               ;PACK ACKNOWLEDGE ERROR
                                        230$:   TST     (SP)+           ;STRIP RETURN TO SUB AND
                                                ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                                                MOVB    220$,@(SP)
                                                BR      400$

                                        ;;**************************************************************************
                                        ;RECALIBRATE DRIVE IF BIT 11 IS SET IN TASK AND "SKI" IS SET
                                        ;OR "PIP" IS ACTIVE.
                                        240$:
                                                BIT     #BIT11,500$     ;RECALIBRATE??
                                                BEQ     410$            ;NO!.

                                                JSR     PC,GET          ;GO GET RMDS
                                                BR      270$            ;NO ERROR GETTING RMDS
                                                BR      260$
                                        250$:   .WORD   0               ;ERROR FROM GET SUB
```

```
040032  062716  000004          260$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040036  113776  040030  000000          MOVB    250$,@(SP)
040044  000470                          BR      400$

040046  032737  040000  040240  270$:   BIT     #SKI,505$       ;WAS SKI SET ?
040054  001004                          BNE     280$            ;YES, GO RECALIBRATE
040056  032737  020000  001350          BIT     #PIP,RMDSI      ;IS PIP ACTIVE??
040064  001462                          BEQ     410$            ;NO.!

040066  012737  000007  001412  280$:   MOV     #RECAL!GO,RMCS10        ;LOAD RECALIBRATE COMMAND
040074  112737  000000  001553          MOVB    #RMCS1,PUTINX   ;AND REGISTER INDEX
040102  112737  000200  001554          MOVB    #200,PUTINX+1
040110  004737  043330                  JSR     PC,PUT          ;GO ISSUE RECALIBRATE
040114  000410                          BR      300$            ;NO ERROR
040116  000401                          BR      290$
040120  000000                          .WORD   0               ;ERROR IN REGISTER TRANSFER
040122  062716  000004          290$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040126  113776  040120  000000          MOVB    290$-2,@(SP)
040134  000434                          BR      400$

040136  004737  043672          300$:   JSR     PC,TIMOUT       ;WAIT FOR COMPLETION

                                ;;*************************************************************
                                ;VERIFY RECALIBRATE IF BIT 3 SET IN TASK
040142  032737  000010  040236          BIT     #BIT3,500$      ;VERIFY RECALIBRATE??
040150  001430                          BEQ     410$            ;NO!!

040152  004737  043060                  JSR     PC,GET          ;GO GET STATUS
040156  000410                          BR      330$            ;NO ERROR GETTING STATUS
040160  000401                          BR      320$
040162  000000                  310$:   .WORD   0               ;ERROR FROM GET
040164  062716  000004          320$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040170  113776  040162  000000          MOVB    310$,@(SP)
040176  000413                          BR      400$

040200  004737  054226          330$:   JSR     PC,RCLSTS       ;GO CHECK RECALIBRATE
040204  000412                          BR      410$            ;NO ERROR DURING RECALIBRATE
040206  000401                          BR      350$
040210  000000                  340$:   .WORD   0               ;ERROR DURING RECALIBRATE
040212  005726                  350$:   TST     (SP)+           ;STRIP RETURN TO SUB AND
040214  062716  000004                  ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040220  113776  040210  000000          MOVB    340$,@(SP)
040226  162716  000002          400$:   SUB     #2,(SP)         ;MOVE SP BACK BEFORE ERROR
040232  000240                  410$:   NOP
040234  000207                          RTS     PC              ;RETURN TO USER

040236  000000                  500$:   .WORD   0               ;TASK/VERIFY DESCRIPTOR
040240  000000                  505$:   .WORD   0               ;CONTAINS RMER2
```

```
                        .SBTTL  BAD SECTOR MODULE
                        ;
                        ;THE MODULE IS INTENDED TO BE CALLED PRIOR TO CALLING THE BUFFER
                        ;GENERATOR SUBROUTINE, AND PRESERVES THE 'PUT BUFFER'' SO THAT THE
                        ;BUFFER NEED ONLY BE FILLED ONCE FOR THE EXECUTION OF A FORMAT
                        ;OPERATION.
                        ;
                        ;THE MODULE RETURNS TO THE CALLING TEST WITH THE APPROVED OR ASSIGNED
                        ;SECTOR IN THE PUT BUFFER AND ALSO IN LOCATIONS ''ASNDA'' AND ''ASNDC''
                        ;SO THAT A REFERENCE IS AVAILABLE TO THE TEST OUTSIDE OF THE PUT BUFFER.
                        ;
                        ;THE BAD SECTOR MODULE PERFORMS TWO MAJOR FUNCTIONS:
                        ;  (1)  RECOVER THE BAD SECTOR FILES AND
                        ;  (2)  APPROVE THE USAGE OF A SECTOR BASED ON INFORMATION IN
                        ;       THE BAD SECTOR FILES OR ASSIGN A NEW SECTOR IF THE ONE
                        ;       ELECTED IS NOT AVAILABLE FOR USE.
                        ;
                        ;INFORMATION REQUIRED BY THE MODULE INCLUDES:
                        ;  (1)  .RMDCO - THE DESIRED CYLINDER,
                        ;  (2)  .RMDAO - THE TRACK AND SECTOR ADDRESS,
                        ;  (3)  .RMWCO - THE WORD COUNT,
                        ;  (4)  .RMCS1O - THE COMMAND,
                        ;  (5)  .RMOFO - THE FORMAT MODE
                        ;
                        ;CALL:
                        ;       JSR     PC,BADSCT       ;CALL SUBROUTINE
                        ;       BR      ???             ;RETURN HERE IF NO ERROR
                        ;       TYPE    ,MESSAGE        ;RETURN HERE IF THE BAD SECTOR FILE
                        ;                               ;CANNOT BE RECOVERED.
                        ;       ERROR   N               ;THE EMT OFFSET NUMBER 'N' IS DEFINED
                        ;                               ;BY BAD SECTOR MODULE.
                        ;
040242                  BADSCT:
040242  062716  000006          ADD     #6,(SP)         ;CLEAR ERROR NUMBER IN USER'S
040246  105076  000000          CLRB    @(SP)           ;ERROR CALL.
040252  162716  000006          SUB     #6,(SP)
                        ;TEST 'MEDIA ENABLE'' TO DETERMINE WHETHER OR NOT THE BAD SECTOR FILES
                        ;HAVE BEEN RECOVERED.
040256  005737  001512          TST     MEDENB          ;HAS BAD SECTOR FILES BEEN RECOVERED ?
040262  001402                  BEQ     1$              ;BR IF NO
040264  000137  041542          JMP     54$             ;YES, BAD SECTOR FILE IS AVAILABLE
                        ;RECOVER THE MANUFACTURES / USERS BAD SECTOR FILE FROM CYLINDER = 822.,
                        ;TRACK   LAST TRACK (RM02/3 = 4 AND RM05 = 18.). ALSO, SAVE THE USER'S
                        ;PUT BUFFER
040270                  1$:
040270  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040272  005000                  CLR     R0              ;START WITH RMCS1
040274  016060  001412  106436  2$:     MOV     PUTBUF(R0),BUFFER(R0)
040302  062700  000002          ADD     #2,R0           ;ADVANCE TO NEXT BUFFER POSITION
040306  022700  000046          CMP     #46,R0          ;END OF BUFFER
040312  103370                  BHIS    2$              ;NO !!
                        ;SET RETRY COUNT AND LOAD PUT BUFFER AND REGISTER INDEX TABLE
                        ;SETUP PARAMETERS TO READ SKIP SECTOR FILE FIRST (3 SECTORS)
040314  012737  000003  042160  MOV     #3,68$          ;RETRY COUNT
```

```
                                                ;DESIRED CYLINDER = 822.
                                                ;STARTING LAST TRACK, SECTOR = 0
                                                ;2 + 256. WORDS (2'S COMP)
                                                ;16 BIT FORMAT
                                                ;POINT TO MANUFACTURES FILE BUFFER

                                                ;R0 POINTS TO REGISTER INDEX TABLE

                                                ;;POP STACK INTO R0

                                ;SET GET INDEX TABLE FOR READING STATUS
                                                ;SETUP GET INDEX REGISTER FOR STATUS
                                                ;GET REGISTERS
                                                ;BR IF NO ERROR
                                                ;JUMP OVER ERROR NUMBER
                                                ;ERROR DEFINED BY GET SUB
                                                ;XFER ERROR TO USER AND
                                                ;GET ERROR NUMBER.

                                                ;GET RMER2 AND SAVE FOR LATER

                                ;CLEAR THE DEVICE USING DRIVE CLEAR COMMAND
                                                ;LOAD COMMAND IN PUT BUFFER
                                                ;OUTPUT COMMAND
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR #
                                                ;ERROR # GOES HERE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR NUMBER TO USER

                                                ;WAIT FOR COMPLETION
                                                ;GO GET STATUS
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR #
                                                ;ERROR # GOES HERE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR # TO USERS ERROR CALL

                                                ;GO VERIFY DRIVE CLEAR COMMAND
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR
                                                ;ERROR # GOES HERE
                                                ;STRIP RETURN TO SUBROUTINE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR # TO USER CALL
```

```
                                        ;ISSUE A PACK ACKNOWLEDGE IF VOLUME VALID IS RESET
040610                                  14$:
040610  032737  000100  001350                BIT     #VV,RMDSI       ;IS VV RESET ??
040616  001052                                BNE     23$             ;NO ..

040620  012737  000023  001412                MOV     #PACACK.GO,RMCS10       ;LOAD COMMAND
040626  004737  043330                        JSR     PC,PUT          ;GO PUT COMMAND TO DRIVE
040632  000411                                BR      17$             ;RETURN HERE IF NO OUTPUT ERROR
040634  000401                                BR      16$             ;GET AROUND ERROR #
040636  000000                          15$:  .WORD   0               ;ERROR # GOES HERE
040640  062716  000006                  16$:  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
040644  113776  040636  000000                MOVB    15$,@(SP)       ;MOVE ERROR # TO ERROR CALL
040652  000137  041246                        JMP     42$

040656  004737  043672                  17$:  JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
040662  004737  043060                        JSR     PC,GET          ;GO GET STATUS FOR PACK ACK
040666  000411                                BR      20$             ;RETURN HERE IF NO ERROR
040670  000401                                BR      19$             ;GET AROUND ERROR #
040672  000000                          18$:  .WORD   0               ;ERROR # GOES HERE
040674  062716  000006                  19$:  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
040700  113776  040672  000000                MOVB    18$,@(SP)       ;MOVE ERROR # TO CALL
040706  000137  041246                        JMP     42$

040712  004737  053432                  20$:  JSR     PC,ACKSTS       ;GO VERIFY ACKNOWLEDGE STATUS
040716  000412                                BR      23$             ;RETURN HERE IF NO ERROR
040720  000401                                BR      22$             ;GET AROUND ERROR #
040722  000000                          21$:  .WORD   0               ;ERROR # GOES HERE
040724  005726                          22$:  TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
040726  062716  000006                        ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
040732  113776  040722  000000                MOVB    21$,@(SP)       ;MOVE ERROR # TO USERS ERROR CALL
040740  000137  041246                        JMP     42$

                                        ;RECALIBRATE THE DRIVE IF ''SKI'' OR 'PIP IS SET
040744                                  23$:
040744  032737  040000  042172                BIT     #SKI,73$        ;WAS SKI SET ?
040752  001004                                BNE     24$             ;YES, GO RECALIBRATE
040754  032737  020000  001350                BIT     #PIP,RMDSI      ;IS PIP SET ??
040762  001452                                BEQ     32$             ;NO !!

040764  012737  000007  001412          24$:  MOV     #RECAL!GO,RMCS10        ;LOAD RECALIBRATE COMMAND
040772  004737  043330                        JSR     PC,PUT          ;PUT THE RECAL COMMAND
040776  000411                                BR      26$             ;RETURN HERE IF NO ERROR
041000  000401                                BR      25$             ;GET AROUND ERROR #
041002  000000                                .WORD   0               ;ERROR # GOES HERE
041004  062716  000006                  25$:  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
041010  113776  041002  000000                MOVB    25$-2,@(SP)     ;MOVE ERROR # TO USERS CALL
041016  000137  041246                        JMP     42$

041022  004737  043672                  26$:  JSR     PC,TIMOUT       ;WAIT FOR RECALIBRATE TO COMPLETE
041026  004737  043060                        JSR     PC,GET          ;GO GET RECAL STATUS
041032  000411                                BR      29$             ;RETURN HERE IF NO ERROR
041034  000401                                BR      28$             ;GET AROUND ERROR #
041036  000000                          27$:  .WORD   0               ;ERROR # GOES HERE
041040  062716  000006                  28$:  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
041044  113776  041036  000000                MOVB    27$,@(SP)       ;MOVE ERROR TO USERS CALL
041052  000137  041246                        JMP     42$
```

```
                                                ;GO VERIFY RECALIBRATE STATUS
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR #
                                                ;ERROR # GOES HERE
                                                ;STRIP RETURN TO SUBROUTINE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR # TO USERS CALL
                                                ;READ THE SECTOR IDENTIFIED BY RMDAO, INCLUDING HEADER AND DATA
                                                ;LOAD READ HEADER AND DATA COMMAND
                                                ;PUT COMMAND
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR #
                                                ;ERROR # GOES HERE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR # TO USERS ERROR CALL
                                                ;WAIT FOR READ OPERATION TO COMPLETE
                                                ;GO GET STATUS FOR READ OPERATION
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR #
                                                ;ERROR # GOES HERE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR # TO CALL
                                                ;GO VERIFY RESULTS OF READ OPERATION
                                                ;RETURN HERE IF NO ERROR
                                                ;GET AROUND ERROR #
                                                ;ERROR # GOES HERE
                                                ;STRIP RETURN ADDRESS TO SUBROUTINE
                                                ;MOVE SP TO USERS ERROR CALL
                                                ;MOVE ERROR # TO USERS CALL
                                                ;ANY CONTROLLER ERRORS ?
                                                ;BR IF YES
                                                ;NO ERRORS DETECTED
                                                ;;*****************************************************
                                                ;AN ERROR HAS BEEN DETECTED IN TRYING TO READ THE BAD SECTOR FILE.
                                                ;THE SECTOR WILL BE RETRIED IF POSSIBLE.
                                                ;YES, DECREMENT RETRY COUNT AND
                                                ;RETRY IF COUNT NOT NEGATIVE.
                                                ;THE RETRY COUNT HAS EXPIRED - SEE IF THE ERROR IS MEDIA RELATED
                                                ;ANY MEDIA RELATED ERRORS ?
                                                ;YES, GO TRY NEXT AVAILABLE SECTOR
                                                ;ANY MEDIA RELATED ERRORS ?
                                                ;NO, EXIT AND REPORT ERROR ON RETURN
```

```
                                        ;THE ERRORS DETECTED WHILE TRYING TO RECOVER THE BAD SECTOR FILE ARE
                                        ;DUE TO THE MEDIA.  SEE IF THE BAD SECTOR FILE CAN BE RECOVERED FROM
                                        ;ANOTHER AREA ON THE LAST TRACK

041274  062737  000002  001420  44$:    ADD     #2,RMDAO        ;ADVANCE SECTOR ADDRESS BY 2
041302  122737  000012  001420          CMPB    #10.,RMDAO      ;QUIT IF ALL MFG SECTORS HAVE BEEN
041310  001413                          BEQ     46$             ;TRIED.
041312  122737  000040  001420          CMPB    #32.,RMDAO      ;QUIT IF ALL USER SECTORS HAVE BEEN
041320  001407                          BEQ     46$             ;TRIED.

041322  012737  000003  042160          MOV     #3,68$          ;REINSTATE RETRY COUNT FOR THIS SECTOR
041330  162716  000006          45$:    SUB     #6,(SP)         ;MOVE SP BACK TO NO ERROR
041334  000137  040422                  JMP     3$              ;RETRY THE READ OPERATION

                                        ;THE BAD SECTOR FILE CANNOT BE READ

041340  000240                  46$:    NOP
041342  032777  020000  137604          BIT     #SW13,@SWR      ;INHIBIT MESSAGE ?
041350  001002                          BNE     47$             ;YES
041352  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO ERROR RETURN
041356  000137  042154          47$:    JMP     67$             ;GO TO MODULE EXIT

                                        ;THE SECTOR WAS RECOVERED WITHOUT ERROR - READ THE USER FILE IF
                                        ;THIS IS THE MGF FILE OR ELSE DONE.

041362  022737  111454  001416  48$:    CMP     #USRFIL,RMBAO   ;WAS THE USER FILE READ ??
041370  001446                          BEQ     52$             ;YES - READ IS COMPLETE
041372  112737  000012  001420          MOVB    #10.,RMDAO      ;READ THE USER FILE LAST TRACK, SECTOR   10.
041400  012737  111454  001416          MOV     #USRFIL,RMBAO   ;POINT TO USERS FILE BUFFER

041406  012737  000003  042160          MOV     #3,68$          ;RELOAD THE RETRY COUNT FOR THIS SECTOR
041414  000137  040422                  JMP     3$              ;GO READ THE USER FILE

                                ;DUMMY THE BAD SECTOR FILES
041420                          49$:
041420  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
041422  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041424  012701  000374                  MOV     #252.,R1        ;R1 = NUMBER OF ENTRIES IN FILES
041430  012700  000014                  MOV     #14,R0          ;R0 = ADDRESS INDEX TO FILE  STORAGE
041434  012760  177777  110446  50$:    MOV     #-1,MFGFIL(R0)  ;ENTER ALL ONES IN MFG FILE
041442  012760  177777  111454          MOV     #-1,USRFIL(R0)  ;ENTER ALL ONES IN USER FILE
041450  005720                          TST     (R0)+           ;ADVANCE ADDRESS
041452  005301                          DEC     R1              ;DECREMENT COUNT
041454  001367                          BNE     50$             ;CONTINUE IF NOT DONE

041456  012701  000006                  MOV     #6.,R1          ;CLEAR HEADER, CLEAR ID & SERIAL NUMBERS
041462  005000                          CLR     R0
041464  005060  110446          51$:    CLR     MFGFIL(R0)
041470  005060  111454                  CLR     USRFIL(R0)
041474  005720                          TST     (R0)+           ;ADVANCE ADDRESS
041476  005301                          DEC     R1
041500  001371                          BNE     51$
041502  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
041504  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0

                                ;SET MEDIA ENABLE AND RESTORE THE USERS PUT BUFFER
041506                          52$.
```

```
     041506  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
282  041510  005000                          CLR     R0              ;R0 IS REGISTER INDEX
283  041512  012737  177777  001512          MOV     #-1,MEDEMB
284  041520  016060  106436  001412  53$:    MOV     BUFFER(R0),PUTBUF(R0)
285  041526  062700  000002                  ADD     #2,R0           ;ADVANCE R0
286  041532  022700  000046                  CMP     #46,R0          ;DONE ??
287  041536  103370                          BHIS    53$
288  041540  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
289
290                                  ;;*****************************************************************
291                                  ;VERIFY THAT THE DESIRED SECTOR IS NOT IN THE MFG BAD SECTOR FILE
292                                  ;AND NOT IN THE USERS BAD SECTOR FILE. ASSIGN A NEW SECTOR IF THE
293                                  ;SECTOR IS IN ANY OF THE FILES.
294                                  ;;*****************************************************************
295
296                                  ;LOAD INITIAL VARIABLES AND COMPUTE THE NUMBER OF SECTORS
297  041542                          54$:
     041542  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
     041544  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
     041546  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
298  041550  013737  001446  001514          MOV     RMDCO,ASNDC     ;LOAD REQUESTED CYLINDER, TRACK,
299  041556  013737  001420  001516          MOV     RMDAO,ASNDA     ;AND SECTOR ADDRESS IN ASSIGNED STORAGE
300  041564  005002                          CLR     R2              ;R2 = NUMBER OF SECTORS
301  041566  013700  001414                  MOV     RMWCO,R0        ;R0 = WORD COUNT
302  041572  005400                          NEG     R0              ;MAKE NUMBER POSITIVE
303  041574  012701  000400                  MOV     #256.,R1        ;R1 = NUMBER OF WORDS PER SECTOR
304  041600  032737  000002  001412          BIT     #BIT1,RMCS10    ;IS THIS A HEADER AND DATA COMMAND ??
305  041606  001402                          BEQ     55$             ;NO !!
306  041610  012701  000402                  MOV     #258.,R1        ;CHANGE WORDS PER SECTOR
307  041614  020100                  55$:    CMP     R1,R0           ;IS THERE A FULL SECTOR ??
308  041616  101404                          BLOS    56$             ;YES !!
309  041620  005700                          TST     R0              ;IS R0 ZERO ??
310  041622  001405                          BEQ     57$             ;YES !!
311  041624  005202                          INC     R2              ;INCREMENT FOR PARTIAL SECTOR
312  041626  000403                          BR      57$
313  041630  160100                  56$:    SUB     R1,R0           ;SUBTRACT ONE SECTOR FROM WORD COUNT
314  041632  005202                          INC     R2              ;INCREMENT SECTOR COUNT
315  041634  000767                          BR      55$
316  041636  010237  042160          57$:    MOV     R2,68$          ;SAVE SECTOR COUNT
317
318                                  ;LOAD PARAMETERS AND SEARCH THE MFG/USER SECTOR FILE FOR THE
319                                  ;ASSIGNED SECTOR. ALSO, SEARCH THE ADJACENT SECTORS IF THE
320                                  ;SECTOR COUNT IS MORE THAN ONE.
321
322  041642  012737  110462  042170          MOV     #MFGFIL+14,72$  ;THE STARTING ADDRESS OF MFG FILE
323                                                                  ;TO BASE ADDRESS STORAGE.
324  041650  004737  041672                  JSR     PC,58$          ;GO SEARCH FILE
325  041654  012737  111470  042170          MOV     #USRFIL+14,72$  ;LOAD STARTING ADDRESS OF USR FILE
326                                                                  ;TO BASE ADDRESS STORAGE.
327  041662  004737  041672                  JSR     PC,58$          ;GO SEARCH FILE
328  041666  000137  042132                  JMP     66$             ;DONE WITH ALL FILE SEARCHES .!
329
330  041672  013737  001514  042164  58$:    MOV     ASNDC,70$       ;LOAD COMPARING CYLINDER ADDRESS
331  041700  013737  001516  042166          MOV     ASNDA,71$       ;LOAD COMPARING TRACK, SECTOR ADDRESS
332  041706  013737  042160  042162          MOV     68$,69$ ;LOAD NUMBER OF SECTORS TO CONFIRM
333
334                                  ;SETUP FOR A BINARY SEARCH OF THE CURRENT FILE FOR THE COMPARING
```

```
                                        ;CYLINDER, TRACK AND SECTOR ADDRESS
041714                                  59$:
041714  013700  042170                          MOV     72$,R0          ;LOAD THE BASE ADDRESS IN R0
041720  022710  177777                  60$:    CMP     #-1,(R0)        ;IS THIS FILE TERMINATOR ?
041724  001446                                  BEQ     64$             ;BR IF YES
041726  021037  042164                          CMP     (R0),70$        ;DOES TABLE ENTRY = COMPARING CYLINDER ?
041732  001010                                  BNE     62$             ;BR IF NO

                                        ;FILE ENTRY EQUALS COMPARING CYLINDER.  SEE IF THE NEXT ENTRY EQUALS
                                        ;THE COMPARING TRACK, AND SECTOR.
041734                                  61$:
041734  126037  000003  042167                  CMPB    3(R0),71$+1     ;DOES TABLE ENTRY = COMPARING TRACK ?
041742  001004                                  BNE     62$             ;BR IF NO

041744  126037  000002  042166                  CMPB    2(R0),71$       ;DOES TABLE ENTRY - COMPARING SECTOR ?
041752  001402                                  BEQ     63$             ;BR IF YES

041754  022020                          62$:    CMP     (R0)+,(R0)+     ;NO, ADJUST CYLINDER POINTER IN BAD FILE
041756  000760                                  BR      60$             ;AND CONTINUE SEARCH.

                                        ;THE COMPARING CYLINDER, TRACK AND SECTOR IS IN THE BAD SECTOR FILE.
                                        ;ADVANCE THE ASSIGNED SECTOR AND START THE SEARCH ALL OVER.
041760                                  63$:
041760  105237  001516                          INCB    ASNDA           ;INCREMENT SECTOR
041764  122737  000037  001516                  CMPB    #31.,ASNDA      ;SECTOR OK ??
041772  103337                                  BHIS    58$             ;YES !!
041774  105037  001516                          CLRB    ASNDA           ;CLEAR SECTOR AND ADVANCE TRACK
042000  105237  001517                          INCB    ASNDA+1
042004  123737  001335  001517                  CMPB    LSTRK+1,ASNDA+1 ;TRACK OK ?
042012  103327                                  BHIS    58$             ;YES !!
042014  005037  001516                          CLR     ASNDA           ;CLEAR TRACK AND SECTOR
042020  005237  001514                          INC     ASNDC           ;INCREMENT CYLINDER
042024  022737  001466  001514                  CMP     #822.,ASNDC     ;CYLINDER OK ??
042032  103317                                  BHIS    58$             ;YES !.
042034  005037  001514                          CLR     ASNDC           ;START AT CYLINDER 0
042040  000714                                  BR      58$             ;SEARCH NEXT SECTOR

                                        ;THE COMPARING SECTOR IS NOT IN THE BAD SECTOR FILES.  DECREMENT THE
                                        ;NUMBER OF SECTORS TO COMPARE AND SEARCH THE NEXT SECTOR IF THE NUMBER
                                        ;IS NOT ZERO.
042042                                  64$:
042042  005337  042162                          DEC     69$             ;DECREMENT NUMBER OF SECTORS TO COMPARE
042046  001442                                  BEQ     67$             ;DONE IF ZERO

042050  105237  042166                          INCB    71$             ;INCREMENT THE COMPARING SECTOR
042054  122737  000037  042166                  CMPB    #31.,71$        ;SECTOR OK ??
042062  103022                                  BHIS    65$             ;YES !!
042064  105037  042166                          CLRB    71$             ;CLEAR SECTOR
042070  105237  042167                          INCB    71$+1           ;INCREMENT TRACK
042074  123737  001335  042167                  CMPB    LSTRK+1,71$+1   ;TRACK OK ??
042102  103012                                  BHIS    65$             ;YES !!
042104  005037  042166                          CLR     71$             ;CLEAR SECTOR TRACK
042110  005237  042164                          INC     70$             ;INCREMENT CYLINDER
042114  022737  001466  042164                  CMP     #822.,70$       ;CYLINDER OK ??
042122  103002                                  BHIS    65$             ;YES !!
042124  005037  042164                          CLR     70$             ;START AT CYLINDER 0
```

```
392 042130  000671                    65$:    BR      59$             ;SEARCH NEXT SECTOR
393
394                                     ;ASSIGN THE SECTOR AND RETURN TO USER
395 042132                              66$:
396 042132  013737  001514  001446              MOV     ASNDC,RMDCO     ;LOAD CYLINDER
397 042140  013737  001516  001420              MOV     ASNDA,RMDAO     ;LOAD TRACK AND SECTOR
398 042146  012602                              MOV     (SP)+,R2        ;;POP STACK INTO R2
    042150  012601                              MOV     (SP)+,R1        ;;POP STACK INTO R1
    042152  012600                              MOV     (SP)+,R0        ;;POP STACK INTO R0
399
400 042154  000240                    67$:    NOP
401 042156  000207                            RTS     PC
402
403                                     ;THE FOLLOWING ARE STORAGE LOCATIONS FOR THE MODULE
404
405 042160  000000                    68$:    .WORD   0       ;RETRY COUNT/ NUMBER OF SECTORS REQUIRED
406 042162  000000                    69$:    .WORD   0       ;NUMBER OF SECTORS TO COMPARE
407 042164  000000                    70$:    .WORD   0       ;COMPARING CYLINDER
408 042166  000000                    71$:    .WORD   0       ;COMPARING TRACK AND SECTOR
409 042170  000000                    72$:    .WORD   0       ;BASE ADDRESS OF BAD SECTOR FILE BEING SEARCHED
410 042172  000000                    73$:    .WORD   0       ;CONTAINS RMER2
```

```
                        .SBTTL  BUFFER GENERATOR SUBROUTINE
                        ;
                        ;THIS SUBROUTINE GENERATES A DATA BUFFER FOR WRITE COMMANDS.  THE
                        ;BUFFER STARTS AT RMBA AND IS RMWC WORDS LONG.  THE BUFFER
                        ;CONTAINS A REPETITIVE DATA PATTERN CONSISTING OF $TMP1 WORDS
                        ;FROM THE DATA PATTERN TABLE, BEGINNING AT ADDRESS $TMP0.
                        ;HEADER INFORMATION FOR THE BUFFER IS EXTRACTED FROM RMDC,
                        ;RMDA AND RMOF.
                        ;
                        ;R0 = ADDRESS OF DATA BUFFER
                        ;R1 - LENGTH OF DATA BUFFER
                        ;R2   ADDRESS OF DATA PATTERN
                        ;R3 = LENGTH OF DATA PATTERN
                        ;R4 = SECTOR COUNT

                        ;CALL:
                        ;(1)    JSR     PC,GENBUF
                        ;(2)    ??                      RETURN HERE

042174                  GENBUF:
042174  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042176  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042200  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042202  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
042204  010446                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
042206  013700  001416         MOV     RMBA0,R0        ;LOAD DATA BUFFER ADDRESS
042212  013701  001414         MOV     RMWC0,R1        ;LOAD WORD COUNT
042216  013737  001446  042426 MOV     RMDC0,60$       ;LOAD STARTING CYLINDER ADDRESS
042224  013737  001420  042430 MOV     RMDA0,65$       ;LOAD STARTING TRACK,SECTOR ADDRESS

042232  032737  000002  001412 10$:    BIT     #BIT1,RMCS10    ;WRITE HEADER & DATA??
042240  001445                 BEQ     25$             ;NO!!
042242  013710  042426         MOV     60$,(R0)        ;WRITE HEADER WORD #1
042246  052710  140000         BIS     #MSE!USE,(R0)   ;SET BAD SECTOR FLAGS FOR GOOD SECTOR
042252  012702  000035         MOV     #29.,R2         ;R2 = MAXIMUM SECTOR ADDRESS (29.)

042256  032737  010000  001444 BIT     #FMT16,RMOF0    ;18 BIT FORMAT??
042264  001404                 BEQ     15$             ;YES !!
042266  052710  010000         BIS     #FMT16,(R0)     ;SET 16 FORMAT BIT IN HEADER
042272  012702  000037         MOV     #31.,R2         ;CHANGE MAXIMUM SECTOR ADDRESS (31.)

042276  005201          15$:    INC     R1              ;INCREMENT WORD COUNT
042300  001443                 BEQ     50$             ;EXIT IF DONE

042302  005720                 TST     (R0)+           ;MOVE R0 TO HEADER WORD #2
042304  013720  042430         MOV     65$,(R0)+       ;WRITE HEADER WORD #2
042310  005201                 INC     R1              ;INCREMENT WORD COUNT AND
042312  001436                 BEQ     50$             ;EXIT IF DONE
042314  012703  042430         MOV     #65$,R3         ;ADVANCE SECTOR ADDRESS
042320  105213                 INCB    (R3)
042322  120213                 CMPB    R2,(R3)         ;SECTOR OVERFLOW ??
042324  103013                 BHIS    25$             ;NO !!
042326  105013                 CLRB    (R3)            ;YES - CLEAR SECTOR ADDRESS
042330  105263  000001         INCB    1(R3)           ;ADVANCE TRACK ADDRESS
042334  123763  001335  000001 CMPB    LSTRK+1,1(R3)   ;TRACK OVERFLOW ??
042342  103004                 BHIS    25$             ;NO !.
042344  105063  000001         CLRB    1(R3)           ;YES - CLEAR TRACK ADDRESS
```

```
                                                INCB    60$             ;ADVANCE CYLINDER ADDRESS
                                        25$:    MOV     #256.,R4        ;LOAD SECTOR DATA COUNT
                                        30$:    MOV     $TMP0,R2        ;LOAD PATTERN ADDRESS
                                                MOV     $TMP1,R3        ;LOAD PATTERN COUNT
                                        40$:    MOV     (R2)+,(R0)+     ;WRITE DATA PATTERN
                                                INC     R1              ;INCREMENT WORD COUNT AND
                                                BEQ     50$             ;EXIT IF DONE
                                                DEC     R4              ;DECREMENT SECTOR COUNT
                                                BEQ     10$             ;START NEXT SECTOR IF 0
                                                DEC     R3              ;DECREMENT PATTERN COUNT
                                                BEQ     30$             ;RESTART PATTERN IF 0
                                                BR      40$             ;CONTINUE DATA PATTERN
                                        50$:
                                                MOV     (SP)+,R4        ;;POP STACK INTO R4
                                                MOV     (SP)+,R3        ;;POP STACK INTO R3
                                                MOV     (SP)+,R2        ;;POP STACK INTO R2
                                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                                MOV     (SP)+,R0        ;;POP STACK INTO R0
                                                NOP
                                                RTS     PC

                                        60$:    .WORD                   ;CYLINDER ADDRESS STORAGE
                                        65$:    .WORD                   ;TRACK, SECTOR ADDRESS STORAGE
```

```
                                        .SBTTL  COMPARE BUFFER SUBROUTINE
                                        ;
                                        ;THIS SUBROUTINE COMPARES THE CONTENTS OF BUFONE AND BUFTWO,
                                        ;ASSUMING THAT BUFONE IS THE BUFFER FROM WHICH DATA WAS WRITTEN
                                        ;AND BUFTWO IS THE BUFFER TO WHICH DATA WAS READ. ERRORS IN BUFFER
                                        ;COMPARISON ARE REPORTED TO THE USER VIA THE USER'S ERROR CALL.
                                        ;
                                        ;CALL:
                                        ;(1)    JSR     PC,CMPBUF
                                        ;(2)    .WORD   WRITE BUFFER ADDRESS
                                        ;       .WORD   READ BUFFER ADDRESS
                                        ;(3)    BR      ??              RETURN HERE IF NO ERROR
                                        ;(4)    NOP                     RETURN HERE IF ERROR
                                        ;(5)    ERROR                   ERROR DEFINED BY SUBROUTINE
                                        ;(6)    ???
                                        ;
042432                                  CMPBUF:
042432  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042434  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042436  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042440  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
042442  005037  042772                        CLR     150$            ;CLEAR CORRECTION FLAG
                                        ;
                                        ;DETERMINE IF DATA SHOULD BE CORRECTED
042446  033737  004000  001370                BIT     ECI,RMOFI       ;WAS ECC CORRECTION ALLOWED ??
042454  001063                               BNE     80$             ;NO !!
042456  032737  100000  001352                BIT     #DCK,RMER1I     ;WAS THERE A DATA CHECK ??
042464  001457                               BEQ     80$             ;NO .!
042466  032737  000100  001352                BIT     #ECH,RMER1I     ;IS ERROR CORRECTION HARD SET ?
042474  001053                               BNE     80$             ;YES !.
042476  032737  010000  001370                BIT     #FMT16,RMOFI    ;IS THIS 16 BIT FORMAT ??
042504  001447                               BEQ     80$             ;NO !!
                                        ;
                                        ;CORRECT DATA USING ECC INFORMATION
042506  013700  001416                        MOV     RMBAO,R0        ;R0 = STARTING BUFFER ADDRESS
042512  013701  001402                        MOV     RMEC1I,R1       ;R1 - ECC POSITION
042516  052737  100000  042772                BIS     #BIT15,150$     ;SET CORRECTION FLAG
                                        ;
                                        ;MOVE R0 TO WORD BOUNDARY OF ERROR BURST
042524  022701  000020                10$:    CMP     #16.,R1         ;IS BIT POSITION > 1 WORD
042530  103004                               BHIS    20$             ;NO !!
042532  162701  000020                        SUB     #16.,R1         ;SUBTRACT 1 WORDS WORTH
042536  005720                               TST     (R0)+           ;ADVANCE BUFFER ADDRESS 1 WORD
042540  000771                               BR      10$
042542  012702  000001                20$:    MOV     #1,R2           ;R2 = BIT POINTER
042546  010203                               MOV     R2,R3           ;R3 = BIT NUMBER
                                        ;
                                        ;MOVE R2 TO STARTING BIT OF ERROR BURST
042550  020301                        30$:    CMP     R3,R1           ;IS R3 SAME AS R1 ??
042552  001403                               BEQ     35$             ;YES !!
042554  006302                               ASL     R2              ;SHIFT BIT POINTER
042556  005203                               INC     R3              ;INCREMENT BIT NUMBER
042560  000773                               BR      30$
042562  012703  000013                35$:    MOV     #11.,R3         ;R3 - LENGTH OF ERROR BURST
                                        ;
                                        ;CORRECT THE ERROR BURST
042566  030237  001404                40$:    BIT     R2,RMEC2I       ;IS THIS BIT SET IN ECC PATTERN ??
```

```
                                                BEQ     60$             ;NO - DO NOT CORRECT THIS BIT
                                                BIT     R2,(R0)         ;IS THE BIT PRESENTLY SET ??
                                                BEQ     50$             ;NO
                                                BIC     R2,(R0)         ;RESET THE BIT
                                                BR      60$
                                        50$:    BIS     R2,(R0)         ;SET THE BIT
                                        60$:    ASL     R2              ;SHIFT TO NEXT BIT
                                                BNE     70$
                                                MOV     #1,R2           ;CONTINUE WITH FIRST BIT OF NEXT WORD
                                                TST     (R0)+
                                        70$:    DEC     R3              ;END OF BURST ??
                                                BNE     40$             ;NO !.

                                        ;COMPARE WRITE BUFFER TO READ BUFFER
                                        80$:    MOV     @10(SP),R0      ;R0 = WRITE BUFFER
                                                ADD     #2,10(SP)       ;MOVE SP TO READ ADDRESS
                                                MOV     @10(SP),R1      ;R1 = READ BUFFER
                                                ADD     #2,10(SP)       ;MOVE SP TO RETURN ADDRESS
                                                MOV     RMWCI,R2        ;R2 = NUMBER OF WORDS TRANSFER
                                                SUB     RMWCO,R2
                                        90$:    CMP     (R0)+,(R1)+     ;COMPARE DATA WORDS
                                                BNE     100$            ;EXIT IF NOT EQUAL
                                                DEC     R2              ;DECREMENT WORD COUNT
                                                BNE     90$             ;CONTINUE IF NOT DONE
                                                BR      110$            ;DONE COMPARE - NO ERROR

                                        ;DATA COMFARE FAILED
                                        100$:   MOV     -(R0),$GDDAT    ;STORE GOOD DATA FOR TYPEOUT
                                                MOV     -(R1),$BDDAT    ;STORE BAD DATA FOR TYPEOUT
                                                MOV     R0,$GDADR       ;STORE ADDRESS OF GOOD DATA
                                                MOV     R1,$BDADR       ;STORE ADDRESS OF BAD DATA
                                                MOV     R2,$TMP0        ;STORE WORD COUNT OF ERROR
                                                ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #336,@10(SP)    ;WRITE ERROR NUMBER IN CALL

                                        ;CHANGE ERROR NUMBER IF ECC CORRECTION FAILED
                                                BIT     #BIT15,150$     ;WAS ECC CORRECTION USED ??
                                                BEQ     105$            ;NO !!
                                                MOVB    #163,@10(SP)    ;ECC CORRECTION FAILED
                                        105$:   SUB     #2,10(SP)       ;MOVE SP TO RETURN IF ERROR
                                                NOP
                                        110$:
                                                MOV     (SP)+,R3        ;;POP STACK INTO R3
                                                MOV     (SP)+,R2        ;;POP STACK INTO R2
                                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                                MOV     (SP)+,R0        ;;POP STACK INTO R0
                                                RTS     PC              ;RETURN TO USER

                                        150$:   .WORD                   ;ECC CORRECTION FLAG
```

```
                        .SBTTL  GET STATUS SUBROUTINE
                        ;
                        ;THIS SUBROUTINE SETS UP THE ''GET INDEX TABLE'' AND THE ''GET
                        ;BUFFER'' FOR READING ALL SUBSYSTEM REGISTERS VIA THE GET SUBROUTINE
                        ;AND THEN RETURNS TO THE USER.
                        ;
                        ;CALL:  JSR     PC,GETSTS
                        ;       ???                             RETURN HERE
                        ;
042774                  GETSTS:
042774  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042776  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043000  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043002  012700  001524          MOV     #GETINX,R0      ;R0 = ADDRESS OF INDEX TABLE
043006  012701  001406          MOV     #RMEC2I+2,R1    ;R1 = ADDRESS OF GET BUFFER
043012  012702  000046          MOV     #RMEC2,R2       ;R2 = REGISTER INDE
043016  110220          2$:     MOVB    R2,(R0)+        ;WRITE REGISTER INDEX IN TABLE
043020  005041                  CLR     -(R1)           ;CLEAR CORRESPONDING LOCATION
043022  162702  000002  3$:     SUB     #2,R2           ;DECREMENT TO NEXT INDEX
043026  100405                  BMI     4$              ;BRANCH OUT IF DONE
043030  022702  000022          CMP     #RMDB,R2        ;DONT WRITE RMDB INDEX
043034  001370                  BNE     2$
043036  005041                  CLR     -(R1)
043040  000770                  BR      3$
043042  112720  000200  4$:     MOVB    #200,(R0)+      ;WRITE TERMINATOR
043046  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
043050  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
043052  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
043054  000240                  NOP
043056  000207                  RTS     PC
```

```
                                .SBTTL  GET SUBROUTINE
                                ;
                                ;THIS SUBROUTINE READS THE REGISTERS WHICH ARE LISTED IN THE
                                ;''GET INDEX TABLE'' AND STORES THEIR VALUES IN THE CORRESPONDING
                                ;LOCATION IN THE ''GET REGISTER BUFFER''.  FOR EXAMPLE, AN
                                ;ENTRY OF 04 IN THE TABLE WILL CAUSE THE SUBROUTINE TO
                                ;READ ''RMBA'' AND STORE ITS CONTENTS AT THE LOCATION IN
                                ;THE BUFFER ASSIGNED TO THAT REGISTER.  THE NUMBER OF
                                ;REGISTERS TO BE READ IS VARIABLE FROM 1 TO 22; THE INDEX
                                ;TABLE MUST BE TERMINATED WITH A CONTROL BYTE (200)
                                ;WHICH SHOULD FOLLOW THE LAST ENTRY.
                                ;
                                ;SUBROUTINE CALL:
                                ;(1)     ''GET INDEX TABLE'' HAS BEEN LOADED WITH REGISTER INDEX
                                ;        VALUES AND TERMINATED WITH A CONTROL BYTE
                                ;(2)     ''GET INPUT BUFFER'' IS AVAILABLE FOR USE.  (NOTE THAT
                                ;        UNUSED LOCATIONS, I.E.,ENTRIES IN BUFFER CORRESPONDING
                                ;        TO REGISTERS NOT READ, ARE NOT CHANGED.)
                                ;(3)     JSR     PC,GET
                                ;        BR      ???             RETURN HERE IF NO ERROR FOUND
                                ;        NOP                     RETURN HERE IF ANY ERROR FOUND
                                ;        ERROR                   SUB DEFINES ERROR NUMBER
                                ;        ???
                                ;
                                ;R0 = REGISTER BASE ADDRESS
                                ;R1 = REGISTER ADDRESS
                                ;R2 = BUFFER BASE ADDRESS
                                ;R3 = BUFFER ADDRESS
                                ;R4 = POINTER TO REGISTER INDEX
                                
043060  000240          GET:    NOP
043062  062716  000004          ADD     #4,(SP)         ;CLEAR ERROR NUMBER IN USER'S
043066  105076  000000          CLRB    @(SP)           ;ERROR CALL
043072  162716  000004          SUB     #4,(SP)
043076  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043100  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043102  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043104  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
043106  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
043110  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
043114  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
043120  013700  001276          MOV     $BASE,R0
043124  012702  001336          MOV     #GETBUF,R2
043130  012704  001524          MOV     #GETINX,R4
043134  012737  043242  000004  MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
043142  012737  000300  000006  MOV     #PR6,ERRVEC+2
043150  016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET ''NED'' STATUS
043156  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET ''DVA'' STATUS
043164  032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
043172  001007                  BNE     3$              ;YES!!
043174  062766  000004  000016  ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN USER'S
043202  112776  000112  000016  MOVB    #112,@16(SP)    ;ERROR CALL
043210  000423                  BR      7$
043212  105714          3$:     TSTB    (R4)            ;DONE??
043214  100433                  BMI     9$              ;YES!!
043216  111401                  MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
043220  042701  177700          BIC     #^CIDXMSK,R1    ;CLEAR ANY SIGN EXTENSION
```

```
 52 043224  060001                        ADD     R0,R1
 53 043226  112403                        MOVB    (R4)+,R3        ;R3 = STORAGE ADDRESS FOR REGISTER
 54 043230  042703  177700                BIC     #^CIDXMSK,R3    ;CLEAR ANY SIGN EXTENSION
 55 043234  060203                        ADD     R2,R3
 56 043236  011113                        MOV     (R1),(R3)       ;READ REGISTER
 57 043240  000764                        BR      3$
 58
 59 043242  022626                5$:     CMP     (SP)+,(SP)+     ;RESTORE STACK
 60 043244  062766  000004  000016        ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
 61 043252  112776  000007  000016        MOVB    #7,@16(SP)      ;USER'S ERROR CALL
 62 043260  162766  000002  000016  7$:   SUB     #2,16(SP)
 63 043266  105714                8$:     TSTB    (R4)            ;DONE CLEARING??
 64 043270  100405                        BMI     9$              ;YES.!
 65 043272  005003                        CLR     R3              ;CLEAR REMAINING STORAGE
 66 043274  112403                        MOVB    (R4)+,R3        ;LOCATIONS
 67 043276  060203                        ADD     R2,R3
 68 043300  005013                        CLR     (R3)
 69 043302  000771                        BR      8$
 70 043304                          9$:
    043304  012637  000006                MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
    043310  012637  000004                MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
    043314  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
    043316  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
    043320  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
    043322  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
    043324  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 71 043326  000207                        RTS     PC              ;RETURN
```

```
                        .SBTTL  PUT SUBROUTINE
                        ;
                        ;THIS SUBROUTINE WRITES THE REGISTERS WHICH ARE LISTED IN THE
                        ;'PUT INDEX TABLE' WITH  THE CONTENTS OF THE CORRESPONDING
                        ;LOCATION IN THE 'PUT REGISTER BUFFER'.  THE NUMBER OF
                        ;REGISTERS WRITTEN IS VARIABLE; THE INDEX TABLE MUST
                        ;BE TERMINATED WITH A CONTROL BYTE (200) WHICH SHOULD
                        ;FOLLOW THE LAST ENTRY.
                        ;
                        ;SUBROUTINE CALL:
                        ;
                        ;(1)    'PUT INDEX TABLE' HAS BEEN LOADED WITH INDEX VALUES
                        ;       OF REGISTERS TO BE WRITTEN.
                        ;(2)    'PUT REGISTER BUFFER' CONTAINS CONTENTS OF EACH
                        ;       REGISTER TO BE WRITTEN.
                        ;(3)    JSR     PC,PUT
                        ;       BR      ???             RETURN HERE IF NO ERROR FOUND
                        ;       NOP                     RETURN HERE IF ANY ERROR FOUND
                        ;       ERROR                   SUB DEFINES ERROR NUMBER
                        ;       ???
                        ;
                        ;R0 = REGISTER BASE ADDRESS
                        ;R1 - REGISTER ADDRESS
                        ;R2 - BUFFER BASE ADDRESS
                        ;R3 - BUFFER ADDRESS
                        ;R4 = POINTER TO REGISTER INDEX
                        ;
 043330  000240         PUT:    NOP
 043332  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 043334  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 043336  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 043340  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 043342  010446                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 043344  013746  000004         MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
 043350  013746  000006         MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
 043354  013700  001276         MOV     $BASE,R0
 043360  012702  001412         MOV     #PUTBUF,R2
 043364  012704  001553         MOV     #PUTINX,R4
 043370  012737  043500  000004 MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
 043376  012737  000300  000006 MOV     #PR6,ERRVEC+2
 043404  016037  000010  001174 1$:     MOV     RMCS2(R0),$TMP0 ;GET 'NED' STATUS
 043412  016037  000000  001176         MOV     RMCS1(R0),$TMP1 ;GET 'DVA' STATUS
 043420  032737  004000  001176         BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
 043426  001007                 BNE     3$              ;YES!!
 043430  062766  000004  000016         ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
 043436  112776  000112  000016         MOVB    #112,@16(SP)    ;USER'S ERROR CALL
 043444  000424                 BR      7$
 043446  105714                 3$:     TSTB    (R4)            ;DONE??
 043450  100425                 BMI     9$              ;YES!!
 043452  111401                 MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
 043454  042701  177700         BIC     #^CIDXMSK,R1    ;CLEAR ANY SIGN EXTENSION
 043460  060001                 ADD     R0,R1
 043462  111403                 MOVB    (R4),R3         ;R3 = STORAGE ADDRESS
 043464  042703  177700         BIC     #^CIDXMSK,R3    ;CLEAR ANY SIGN EXTENSION
 043470  060203                 ADD     R2,R3
 043472  011311                 MOV     (R3),(R1)       ;WRITE REGISTER
 043474  105724                 4$:     TSTB    (R4)+           ;ADJUST REGISTER POINTER
```

SEQ 0195

```
     52 043476  000763                          BR      3$
     53
     54 043500  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
     55 043502  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     56 043510  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     57 043516  162766  000002  000016  7$:     SUB     #2,16(SP)
     58
     59 043524                          9$:
        043524  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        043530  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        043534  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        043536  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        043540  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        043542  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        043544  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     60 043546  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  SIZE CLOCK SUBROUTINE
                                        
043550                                  SIZCLK:
043550  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
043554  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
043560  012737  043616  000004                  MOV     #1$,ERRVEC      ;SET UP FOR BUS TIMEOUT
043566  012737  000300  000006                  MOV     #PR6,ERRVEC+2
043574  012737  177546  001520                  MOV     #177546,CLKADR  ;LOAD ADDRESSES FOR KW11-L
043602  012737  000100  001522                  MOV     #100,CLKVCT
043610  005777  135704                          TST     @CLKADR         ;TEST FOR KW11-L PRESENT
043614  000421                                  BR      3$              ;YES - KW11-L IS PRESENT
043616  022626                          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP
043620  012737  043650  000004                  MOV     #2$,ERRVEC      ;SET UP FOR BUS TIMEOUT
043626  012737  172540  001520                  MOV     #172540,CLKADR  ;LOAD ADDRESSES FOR KW11-P CLOCK
043634  012737  000104  001522                  MOV     #104,CLKVCT
043642  005777  135652                          TST     @CLKADR         ;TEST FOR KW11-P PRESENT
043646  000404                                  BR      3$              ;YES - KW11-P IS PRESENT
043650  022626                          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP
043652  062766  000002  000004                  ADD     #2,4(SP)        ;MOVE RETURN TO ERROR
043660                                  3$:
043660  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
043664  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
043670  000207                                  RTS     PC              ;RETURN TO USER
```

```
                        .SBTTL  TIMEOUT SUBROUTINE
                        ;
                        ;THIS SUBROUTINE WAITS FOR RDY   1 AND GO = 0 OR FOR A TIMEOUT
                        ;GREATER THAN APPROX. 500 MSEC AND THEN RETURNS.
                        ;
                        ;CALL:  JSR     PC,TIMOUT
                        ;       ???                     RETURN HERE
                        ;
043672                  TIMOUT:
043672  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043674  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043676  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043700  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
043704  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
043710  012737  044012  000004  MOV     #4$,ERRVEC      ;SETUP FOR BUS TIMEOUT - 04 TRAP
043716  012737  000300  000006  MOV     #PR6,ERRVEC+2
043724  013700  001276          MOV     $BASE,R0        ;R0=BASE ADDRESS
043730  013701  001520          MOV     CLKADR,R1       ;R1=CLOCK ADDRESS
043734  012702  000036          MOV     #30.,R2         ;R2=NUMBER OF CLOCK CYCLES
043740  020127  172540  1$:     CMP     R1,#172540      ;KW11-P CLOCK??
043744  001003                  BNE     2$              ;NO!!
043746  012761  000001  000002  MOV     #1,2(R1)        ;SET COUNTER
043754  012711  000005  2$:     MOV     #BIT2!BIT0,(R1) ;START COUNTER

043760  016046  000000  3$:     MOV     RMCS1(R0),-(SP) ;GET STATUS
043764  042716  177576          BIC     #^C<RDY!GO>,(SP)
043770  022726  000200          CMP     #RDY,(SP)+      ;RDY=1,GO-0??
043774  001420                  BEQ     5$              ;YES!!
043776  032711  000200          BIT     #BIT7,(R1)      ;TIMER DONE??
044002  001766                  BEQ     3$              ;NO!!
044004  005302                  DEC     R2              ;DEC NUMBER OF CYCLES
044006  001354                  BNE     1$              ;CONTINUE IF NOT DONE
044010  000412                  BR      5$              ;'RDY' DID NOT SET OR ''GO'' DID NOT RESET
                                                        ;WITHIN 500 MSEC AFTER THE COMMAND WAS ISSUED.
044012  022626          4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
044014  062766  000004  000012  ADD     #4,12(SP)       ;MOVE SP TO USER'S CALL
044022  112776  000007  000012  MOVB    #7,@12(SP)      ;WRITE ERROR NUMBER
044030  162766  000002  000012  SUB     #2,12(SP)

044036                  5$:
044036  012637  000006          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
044042  012637  000004          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
044046  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
044050  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
044052  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
044054  000207                  RTS     PC              ;RETURN TO USER
```

```
                                .SBTTL  PRIMARY ERROR CHECK SUBROUTINE
                                ;
                                ;THE PURPOSE OF THIS SUBROUTINE IS TO VERIFY THAT STATUS IS VALID AND
                                ;THAT FURTHER ERROR AND STATUS CHECKING SHOULD BE PERFORMED.  THE
                                ;FOLLOWING CHECKS ARE MADE:
                                ;
                                ;       .CORRECT UNIT IS SELECTED, I.E., THE UNIT SELECT BITS OF RMCS2
                                ;(BITS 0-2) EQUAL THE UNIT BEING TESTED;
                                ;
                                ;       .SELECTED UNIT IS AVAILABLE, I.E., DVA (BIT 11 OF RMCS1) IS SET
                                ;AND NED (BIT 12 OF RMCS2) IS RESET;
                                ;
                                ;       .LAST COMMAND WAS COMPLETED, I.E., THE MASSBUS CONTROLLER IS
                                ;READY (BIT 7 OF RMCS1) AND THE GO BIT IS RESET (BIT 0 OF RMCS1) OR THE
                                ;DRIVE READY BIT (BIT 7 OF RMDS) IS SET.
                                ;       .NO PARITY ERROR OCCURRED WHEN READING REMOTE REGISTERS,
                                ;I.E., MCPE = 0.
                                ;       .NO PARITY ERROR OCCURRED WHEN WRITING REMOTE REGISTERS,
                                ;I.E., PAR - 0, OR, PAR = DPE = 1
                                ;
                                ;THE SUBROUTINE ASSUMES THAT:
                                ;
                                ;       .STATUS HAS BEEN STORED IN THE REGISTER INPUT BUFFER,
                                ;IN PARTICULAR, RMCS1, RMCS2 AND RMDS HAVE BEEN STORED IN THEIR
                                ;CORRESPONDING LOCATIONS OF THE ''GET'' BUFFER.
                                ;
                                ;       .($UNIT) CONTAINS THE DRIVE NUMBER
                                ;
                                ;THE SUBROUTINE IS CALLED AS FOLLOWS:
                                ;
                                ;(1)    JSR     PC,PRIERR
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS
                                ;
044056                          PRIERR:
                                ;
                                ;CLEAR USER'S ERROR CALL
044056  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
044062  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
044066  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
                                ;
                                ;REPORT AN ERROR IF THE WRONG UNIT IS SELECTED
044072  013737  001346  001142          MOV     RMCS2I,$BDDAT   ;CORRECT UNIT SELECTED??
044100  042737  177770  001142          BIC     #^CUNTMSK,$BDDAT
044106  013737  001234  001140          MOV     $UNIT,$GDDAT    ;GOOD DATA FOR TYPEOUT
044114  042737  177770  001140          BIC     #^CUNTMSK,$GDDAT
044122  123737  001140  001142          CMPB    $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                                                ;DRIVE NUMBERS
044130  001415                          BEQ     1$              ;YES..
044132  062716  000004                  ADD     #4,(SP)
044136  112776  000001  000000          MOVB    #1,@(SP)        ;ERROR 1
044144  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
044150  004736                          JSR     PC,@(SP)+       ;REPORT WRONG UNIT SELECTED
```

```
     58 044152  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
     59 044156  000240                          NOP
     60 044160  000137  044700                  JMP     10$             ;SKIP OTHER CHECKS
     61 044164                          1$:
     62
     63                                 ;REPORT AN ERROR IF THE DEVICE IS NOT AVAILABLE OR IF
     64                                 ;THE DEVICE IS NONEXISTANT
     65 044164  032737  004000  001336          BIT     #DVA,RMCS1I     ;DEVICE AVAILABLE??
     66 044172  001045                          BNE     5$              ;YES!!
     67 044174  013737  001336  001140          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
     68 044202  052737  004000  001140          BIS     #DVA,$GDDAT
     69 044210  013737  001336  001142          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
     70 044216  062716  000004                  ADD     #4,(SP)
     71 044222  112776  000002  000000          MOVB    #2,@(SP)        ;ERROR #2
     72 044230  032737  010000  001346          BIT     #NED,RMCS2I     ;WAS NED SET??
     73 044236  001414                          BEQ     2$              ;NO!!
     74 044240  013737  001346  001140          MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
     75 044246  013737  001346  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
     76 044254  042737  010000  001140          BIC     #NED,$GDDAT
     77 044262  112776  000003  000000          MOVB    #3,@(SP)        ;YES - CHANGE ERROR NUMBER
     78 044270  162716  000002          2$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     79 044274  004736                          JSR     PC,@(SP)+       ;REPORT DEVICE NOT AVAILABLE
     80 044276  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
     81 044302  000240                          NOP
     82 044304  000575                          BR      10$             ;SKIP OTHER CHECKS
     83 044306                          5$:
     84
     85                                 ;REPORT AN ERROR IF MASSBUS CONTROLLER IS NOT READY
     86 044306  032737  000200  001336          BIT     #RDY,RMCS1I     ;CONTROLLER READY??
     87 044314  001030                          BNE     7$              ;YES!!
     88 044316  013737  001336  001140          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
     89 044324  052737  000200  001140          BIS     #RDY,$GDDAT
     90 044332  042737  160001  001140          BIC     #SC.TRE!MCPE!GO,$GDDAT
     91 044340  013737  001336  001142          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
     92 044346  062716  000004                  ADD     #4,(SP)
     93 044352  112776  000004  000000          MOVB    #4,@(SP)        ;ERROR #4
     94 044360  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     95 044364  004736                          JSR     PC,@(SP)+       ;REPORT CONTROLLER NOT READY
     96 044366  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
     97 044372  000240                          NOP
     98 044374  000541                          BR      10$             ;SKIP OTHER CHECKS
     99 044376                          7$:
    100
    101                                 ;REPORT AN ERROR IF GO IS NOT ZERO AND DRY IS NOT ONE
    102 044376  032737  000001  001336          BIT     #GO,RMCS1I      ;GO RESET??
    103 044404  001431                          BEQ     8$              ;YES!!
    104 044406  032737  000200  001350          BIT     #DRY,RMDSI      ;DRIVE READY??
    105 044414  001025                          BNE     8$              ;YES!!
    106 044416  013737  001336  001140          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
    107 044424  042737  160001  001140          BIC     #SC!TRE!MCPE!GO,$GDDAT
    108 044432  013737  001336  001142          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
    109 044440  062716  000004                  ADD     #4,(SP)
    110 044444  112776  000005  000000          MOVB    #5,@(SP)        ;ERROR #5
    111 044452  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    112 044456  004736                          JSR     PC,@(SP)+       ;REPORT DRIVE NOT READY
    113 044460  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
    114 044464  000240                          NOP
```

```
                                                BR      10$
                                        8$:
                                        ;REPORT AN ERROR IF THE RH CONTROLLER DETECTED BAD
                                        ;PARITY ON THE MASSBUS CONTROL BUS
044470  032737  020000  001336            BIT     #MCPE,RMCS1I    ;PARITY ERROR ??
044476  001425                            BEQ     9$              ;NO!!
044500  013737  001336  001140            MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
044506  042737  160001  001140            BIC     #SC!TRE!MCPE!GO,$GDDAT
044514  013737  001336  001142            MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
044522  062716  000004                    ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
044526  112776  000013  000000            MOVB    #13,@(SP)       ;ERROR #13
044534  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
044540  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
044542  162716  000010                    SUB     #10,(SP)        ;RESTORE STACK
044546  000240                            NOP
044550  000453                            BR      10$
044552                                  9$:
                                        ;REPORT AN ERROR IF DETECTED A CONTROL BUS PARITY ERROR
044552  032737  000010  001352            BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
044560  001451                            BEQ     11$             ;NO!!
044562  032737  000010  001400            BIT     #DPE,RMER2I     ;WAS IT THE CONTROL BUS??
044570  001045                            BNE     11$             ;NOT SURE!!
044572  032737  000010  001426            BIT     #PAR,RMER1O     ;DID TEST SET PAR ??
044600  001413                            BEQ     93$             ;NO!!
044602  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044604  012700  001553                    MOV     #PUTINX,R0      ;R0 POINTS TO INDEX TABLE
044610  122710  000014            91$:    CMPB    #RMER1,(R0)     ;SEARCH TABLE FOR RMER1
044614  001002                            BNE     92$
044616  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
044620  000431                            BR      11$             ;PAR WAS SET BY TEST
044622  105720                    92$:    TSTB    (R0)+           ;END OF TABLE??
044624  100371                            BPL     91$             ;NO!!
044626  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
044630  013737  001352  001140    93$:    MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
044636  042737  000010  001140            BIC     #PAR,$GDDAT
044644  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
044652  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
044656  112776  000050  000000            MOVB    #50,@(SP)       ;WRITE THE ERROR NUMBER
044664  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
044670  004736                            JSR     PC,@(SP)+       ;REPORT THE ERROR
044672  162716  000010                    SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
044676  000240                            NOP
044700  062716  000010            10$:    ADD     #10,(SP)        ;RETURN TO ERROR
044704  000240                    11$:    NOP                     ;RETURN TO NO ERROR
044706  000207                            RTS     PC
```

```
                                .SBTTL  SECONDARY ERROR CHECK SUBROUTINE

                                ;THE ERROR CHECK SUBROUTINE PROVIDES DETECTION OF SECONDARY ERRORS
                                ;SUCH AS UNEXPECTED ERRORS AND UNEXPECTED REGISTER CONTENTS. THESE
                                ;ERRORS ARE DEEMED SECONDARY IN THAT THEY ARE NOT NECESSARILY
                                ;ASSOCIATED WITH THE OPERATION BEING PERFORMED.
                                ;WHEN THE SUBROUTINE IDENTIFIES SUCH AN ERROR, IT MOVES THE ERROR
                                ;NUMBER TO THE ERROR CALL IN THE TEST ROUTINE AND THEN RETURNS
                                ;TO THE TEST ROUTINE WHICH MAKES THE ERROR CALL. AFTER THE TEST ROUTINE
                                ;MAKES THE ERROR CALL, IT RETURNS TO THE SUBROUTINE WHICH THEN LOOKS FOR
                                ;OTHER ERRORS. WHEN ALL ERRORS HAVE BEEN REPORTED, THE SUBROUTINE
                                ;RETURNS TO THE ADDRESS FOLLOWING THE SUBROUTINE CALL.
                                ;
                                ;CALL:  JSR     PC,SECERR
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS

                                ;NOTE:  THE SUBROUTINE ASSUMES THAT REGISTERS HAVE BEEN STORED AT THE
                                ;INPUT REGISTER BUFFER.

044710                          SECERR:

                                ;;*****************************************************************
                                ;STORE FUNCTION CODE AND CLEAR USER'S ERROR NUMBER
044710  013737  001412  050544          MOV     RMCS10,515$     ;STORE FUNCTION CODE
044716  042737  177701  050544          BIC     #^C<F0!F1!F2.F3!F4>,515$
044724  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
044730  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
044734  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

                                ;;*****************************************************************
                                ;CHECK SECONDARY ERRORS COMMON TO ALL COMMANDS

                                ;REPORT ERROR IF DRIVE IS NOT READY, I.E., IF DRY = 0
044740  032737  000200  001350          BIT     #DRY,RMDSI      ;DRIVE READY??
044746  001024                  BNE     5$              ;YES!!
044750  013737  001350  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
044756  042737  177577  001142          BIC     #^CDRY,$BDDAT
044764  012737  000200  001140          MOV     #DRY,$GDDAT     ;GOOD DATA FOR TYPEOUT
044772  062716  000004                  ADD     #4,(SP)
044776  112776  000010  000000          MOVB    #10,@(SP)       ;ERROR NUMBER
045004  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045010  004736                  JSR     PC,@(SP)+       ;REPORT NOT READY
045012  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP) TO ERROR N
045016  000240                  NOP

                                ;REPORT ERROR IF GO BIT IS NOT RESET
045020  032737  000001  001336  5$:     BIT     #GO,RMCS1I      ;GO BIT RESET??
045026  001423                  BEQ     10$             ;YES..
045030  013737  001336  001142          MOV     RMCS1I,$BDDAT   ;BAD DATA FOR TYPEOUT
045036  042737  177776  001142          BIC     #^CGO,$BDDAT
045044  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
045050  062716  000004                  ADD     #4,(SP)
045054  112776  000011  000000          MOVB    #11,@(SP)       ;ERROR NUMBER
```

```
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT DEVICE NOT AVAILABLE
                                        SUB     #10,(SP)                ;RESTORE (SP)
                                        NOP

                                ;REPORT ERROR IF FUNCTION CODE READ FROM DEVICE IS NOT CORRECT
                                10$:    MOV     RMCS1I,$BDDAT   ;IS FUNCTION CODE CORRECT??
                                        BIC     #^C76,$BDDAT
                                        MOV     515$,$GDDAT     ;EXPECTED FUNCTION CODF
                                        CMP     $BDDAT,$GDDAT
                                        BEQ     15$             ;YES.!
                                        ADD     #4,(SP)
                                        MOVB    #12,@(SP)       ;ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT WRONG FUNCTION CODE
                                        SUB     #10,(SP)                ;RESTORE (SP)
                                        NOP
                                15$:
                                ;REPORT AN ERROR IF COMPOSITE ERROR IS SET AND NO OTHER
                                ;ERRORS ARE SET, OR IF COMPOSITE ERROR IS NOT SET AND
                                ;OTHER ERRORS ARE SET
                                        CLR     $GDDAT          ;EXPECT 'ERR' - 0
                                        TST     RMER1I          ;IS RMER1 = 0??
                                        BNE     20$             ;NO!!
                                        TST     RMER2I          ;IS RMERZ = 0??
                                        BEQ     25$             ;YES.!
                                20$:    BIS     #ERR,$GDDAT     ;'ERR' SOULD BE SET
                                25$:    MOV     RMDSI,$BDDAT
                                        BIC     #^CERR,$BDDAT
                                        CMP     $GDDAT,$BDDAT   ;IS 'ERR' OK??
                                        BEQ     30$             ;YES!!
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                        MOVB    #47,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT INVALID COMP ERROR
                                        SUB     #10,(SP)

                                ;REPORT AN ERROR IF 'TRE' IS SET AND NONE OF THE BITS WHICH SET
                                ;TRE IS SET, OR IF TRE IS NOT SET AND ONE OR MORE BITS WHICH
                                ;SET TRE IS SET
                                30$:    CLR     $GDDAT          ;EXPECT 'TRE' - 0
                                        MOV     RMCS2I,-(SP)    ;WAS DLT, WCE, UPE, NED, NEM
                                        BIC     #377,(SP)+      ;PGE, MXF OR MDPE SET
                                        BNE     35$             ;YES!.
                                        BIT     #ERR,RMDSI      ;WAS EXCEPTION RECEIVED??
                                        BEQ     40$             ;NO..
                                        CMP     #SEARCH,515$    ;WAS DATA TRANSFERRED??
                                        BHIS    40$             ;NO.!
                                35$:    BIS     #TRE,$GDDAT     ;'TRE' SHOULD BE SET
                                40$:    MOV     RMCS1I,$BDDAT   ;BAD DATA FOR TYPEOUT
                                        BIC     #^CTRE,$BDDAT
                                        CMP     $GDDAT,$BDDAT   ;IS 'TRE' OK??
                                        BEQ     45$             ;YES.!
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #14,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT TRE ERROR
```

```
045364  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
045370  000240                          NOP
045372                          45$:

                                ;;*****************************************************************
                                ;USING THE FUNCTION CODE TABLE, CHECK FOR THE FOLLOWING ERRORS:
                                ;       .STATUS BITS NOT SET THAT SHOULD BE SET, E.G., ATA AND ILF
                                ;       .STATUS BITS SET THAT SHOULD NOT BE SET, E.G., WCE AND ECH
                                ;NOTE THAT SOME ERROR BITS ARE CONDITIONAL ON THE COMMAND AND OTHER
                                ;STATUS CONDITIONS, E.G., WRITE LOCK ERROR SHOULD ONLY BE SET IF
                                ;WRITE LOCK IS ON AND THE COMMAND IS A WRITE.

                                ;GET AND STORE THE FNTRY FROM THE FUNCTION CODE TABLE
045372  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
045374  013700  050544                  MOV     515$,R0         ;GET FUNCTION CODE
045400  016037  071646  050536          MOV     FNCDTB(R0),500$ ;STORE ENTRY
045406  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0

                                ;REPORT AN ERROR IF AN UNEXPECTED ATTENTION OCCURRED OR IF
                                ;ATA IS NOT SET AND SHOULD BE SET.
045410  013737  050536  001140          MOV     500$,$GDDAT     ;GET EXPECTED ATA STATUS
045416  032737  040000  001350          BIT     #ERR,RMDSI      ;IS COMPOSITE ERROR SET ??
045424  001403                          BEQ     50$             ;NO !!
045426  052737  100000  001140          BIS     #ATA,$GDDAT     ;EXPECT AN ATTENTION
045434  042737  077777  001140  50$:    BIC     #^CATA,$GDDAT   ;STRIP DONT CARES
045442  013737  001350  001142          MOV     RMDSI,$BDDAT    ;GET RECEIVED ATA
045450  042737  077777  001142          BIC     #^CATA,$BDDAT   ;STRIP DONT CARES
045456  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA OK ??
045464  001413                          BEQ     55$             ;YES !!
045466  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
045472  112776  000006  000000          MOVB    #6,@(SP)        ;LOAD ERROR # IN CALL
045500  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
045504  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
045506  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
045512  000240                          NOP
045514                          55$:

                                ;REPORT ERROR IF ILF IS INCORRECT, I.E., IF ILF DOES NOT COMPARE
                                ;WITH FUNCTION CODE TABLE
045514  013737  050536  001140          MOV     500$,$GDDAT     ;GET EXPECTED ILF
045522  042737  177776  001140          BIC     #^CILF,$GDDAT   ;CLEAR ALL OTHER BITS
045530  013737  001352  001142          MOV     RMER1I,$BDDAT   ;GET RECEIVED ILF
045536  042737  177776  001142          BIC     #^CILF,$BDDAT   ;CLEAR ALL OTHER BITS
045544  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ILF OK ??
045552  001412                          BEQ     60$             ;YES !!
045554  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
045560  112776  000254  000000          MOVB    #254,@(SP)      ;WRITE ERROR NUMBER IN CALL
045566  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
045572  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
045574  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
045600  005037  001140          60$:    CLR     $GDDAT          ;CLEAR EXPECTED STATUS

                                ;REPORT AN ERROR IF WCE IS SET AND SHOULD NOT BE SET
045604  013746  050536                  MOV     500$,-(SP)      ;GET WCE STATUS ENABLE
045610  052716  137777                  BIS     #^CWCE,(SP)     ;SET ALL OTHER BITS
045614  013737  001346  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
045622  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR WCE IF ENABLED
```

```
                                        BEQ     90$             ;BRANCH IF WCE OK
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #26,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR
                                        SUB     #10,(SP)        ;RESTORE ERROR
                                90$:

                                ;REPORT ERROR IF OPI STATUS IS SET AND SHOULD NOT BE SET
                                        MOV     500$,-(SP)      ;GET OPI STATUS ENABLE
                                        BIS     #^COPI,(SP)     ;SET ALL OTHER BITS
                                        MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR OPI IF ENABLED
                                        BEQ     100$            ;BRANCH IF OPI OK
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #164,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR
                                        SUB     #10,(SP)        ;RESTORE SP
                                100$:

                                ;REPORT ERROR IF IVC IS SET AND IS NOT ENABLED OR IF IVC IS
                                ;SET AND VV IS NOT RESET
                                        MOV     500$,-(SP)      ;GET IVC STATUS ENABLE
                                        BIT     #VV,RMDSI       ;IS VV SET
                                        BEQ     105$            ;NO !!
                                        BIC     #IVC,(SP)       ;YES - IVC SHOULD BE 0
                                105$:   BIS     #^CIVC,(SP)     ;SET ALL OTHER BITS
                                        MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR IVC IF ENABLED
                                        BEQ     110$            ;BRANCH IF IVC OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #165,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR
                                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                110$:

                                ;BIT 11 (WLE) OF THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR
                                ; ALL WRITE ERRORS, I.E.,
                                ;       RMER1 - WLE, WCF
                                ;       RMER2 - DPE
                                ;       RMCS2 - UPE.
                                ;EACH OF THESE ERRORS IS CHECKED TO SEE IF AN ERROR IS SET WHEN THE
                                ;WRITE ERROR ENABLE BIT IS RESET.

                                ;REPORT AN ERROR IF WLE IS SET AND WRITE ERRORS ARE NOT ENABLED, OR IF
                                ;THE DRIVE IS NOT WRITE PROTECTED
                                        MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
                                        BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                        BEQ     115$            ;NO !.
                                        BIT     #WRL,RMDSI      ;IS THE DRIVE WRITE PROTECTED ??
                                        BNE     120$            ;YES !!
                                115$:   BIC     #WLE,(SP)       ;RESET WLE ENABLE
                                120$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR WLE IF ENABLED
                                        BEQ     125$            ;BRANCH IF WLE OK
```

```
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #23,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        125$:
                                        ;REPORT ERROR IF WCF IS SET AND WRITE ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
                                                BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                                BNE     130$            ;YES !!
                                                BIC     #WCF,(SP)       ;DISABLE WCF ERROR
                                        130$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET WCF IF ENABLED
                                                BEQ     135$            ;BRANCH IF WCF OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #25,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        135$:
                                        ;REPORT ERROR IF DPE IS SET AND WRITE ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
                                                BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                                BNE     140$            ;YES !!
                                                BIC     #DPE,(SP)       ;RESET DPE ENABLE
                                        140$:   MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET DPE IF ENABLED
                                                BEQ     145$            ;BRANCH IF DPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #40,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        145$:
                                        ;REPORT AN ERROR IF UPE IS SET AND WRITE ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
                                                BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                                BNE     150$            ;YES !!
                                                BIC     #UPE,(SP)       ;DISABLE UPE ERROR
                                        150$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET UPE IF ENABLED
                                                BEQ     155$            ;BRANCH IF UPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #24,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        155$:
                                        ;REPORT AN ERROR IF IAE IS SET AND IS NOT ENABLED
                                                MOV     500$,-(SP)      ;GET IAE ENABLE
                                                BIS     #^CIAE,(SP)     ;SET ALL OTHER BITS
                                                MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR IAE IF ENABLED
```

```
                        BEQ     160$            ;BRANCH IF IAE IS OK
                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                        MOVB    #166,@(SP)      ;WRITE ERROR NUMBER
                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                160$:

                ;BIT 09 (AOE) OF THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR
                ; ALL READ/WRITE ERRORS, I.E.,
                ;       RMCS1 - TRE
                ;       RMCS2 - DLT,NEM,MXF
                ;       RMDS  - LBT
                ;       RMER1 - AOE
                ;NOTE:
                ;       LBT IS NOT CHECKED BECAUSE IT ONLY RESETS WHEN THE DESIRED
                ;CYLINDER REGISTER IS WRITTEN
                ;NOTE:
                ;       AOE CANNOT BE SET IF LBT IS NOT ALSO SET
                ;NOTE:
                ;       TRE IS CHECKED AS A FUNCTION OF OTHER ERROR CONDITONS ABOVE


                ;REPORT AN ERROR IF DLT IS SET AND READ/WRITE ERRORS ARE NOT ENABLED
                        MOV     #-1,-(SP)       ;ASSUME ERRORS ARE ENABLED
                        BIT     #AOE,500$       ;ARE ERRORS ENABLED ??
                        BNE     165$            ;YES !!
                        BIC     #DLT,(SP)       ;RESET DLT ENABLE
                165$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                        BIC     (SP)+,$BDDAT    ;CLEAR DLT IF ENABLED
                        BEQ     170$            ;BRANCH IF DLT IS OK
                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                        MOVB    #32,@(SP)       ;WRITE ERROR NUMBER IN CALL
                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                170$:

                ;REPORT ERROR IF NEM IS SET AND READ/WRITE ERRORS ARE NOT ENABLED
                        MOV     #-1,-(SP)       ;ASSUME ERRORS ARE ENABLED
                        BIT     #AOE,500$       ;ARE ERRORS ENABLED ??
                        BNE     175$            ;YES !!
                        BIC     #NEM,(SP)       ;DISABLE NEM
                175$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                        BIC     (SP)+,$BDDAT    ;CLEAR NEM IF ENABLED
                        BEQ     180$            ;BRANCH IF NEM IS OK
                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                        MOVB    #167,@(SP)      ;WRITE ERROR NUMBER IN CALL
                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                180$:

                ;REPORT ERROR IF MXF IS SET AND READ/WRITE ERRORS ARE NOT ENABLED
                        MOV     #-1,-(SP)       ;ASSUME ERRORS ARE ENABLED
                        BIT     #AOE,500$       ;ARE DATA ERRORS ENABLED ??
                        BNE     185$            ;YES !!
```

```
                                        BIC     #MXF,(SP)       ;DISABLE MXF ERROR
                                        MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR MXF IF ENABLED
                                        BEQ     190$            ;BRANCH IF MXF IS OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #33,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                190$:

                                ;REPORT ERROR IF AOE IS SET AND DATA ERRORS ARE NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME DATA ERRORS ARE ENABLED
                                        BIT     #AOE,500$       ;ARE DATA ERRORS EANBLED ??
                                        BEQ     191$            ;NO !!
                                        BIT     #LBT,RMDSI      ;IS LBT ALSO SET ??
                                        BNE     195$            ;YES !!
                                191$:   BIC     #AOE,(SP)       ;DISABLE AOE
                                195$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR AOE IF ENABLED
                                        BEQ     200$            ;BRANCH IF AOE IS OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #20,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                200$:

                                ;BIT 07 (HCE) OF THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR
                                ;HEADER ERRORS, I.E.,
                                ;       RMER1 - HCRC,HCE,FER
                                ;       RMER2 - BSE

                                ;RESET THE ENABLING BIT (HCE) IF HEADER COMPARE INHIBIT IS SET
                                        BIT     #HCI,RMOFI      ;IS HCI SET ??
                                        BEQ     201$            ;NO !.
                                        BIC     #HCE,500$       ;YES - DISABLE ALL HEADER ERRORS
                                201$:

                                ;REPORT AN ERROR IF HCRC IS SET AND HEADER ERRORS ARE NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                        BIT     #HCE,500$       ;ARE HEADER ERRORS ENABLED ??
                                        BNE     205$            ;YES !!
                                        BIC     #HCRC,(SP)      ;DISABLE HCRC
                                205$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;RESET HCRC IF ENABLED
                                        BEQ     210$            ;BRANCH IF HCRC IS OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #35,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                210$:

                                ;REPORT ERROR IF HCE IS SET AND HEADER ERRORS ARE NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                        BIT     #HCE,500$       ;ARE ERRORS ENABLED ??
```

```
                                                BNE     215$            ;YES !!
                                                BIC     #HCE,(SP)       ;DISABLE HCE
                                        001142  215$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;CLEAR HCE IF ENABLED
                                                        BEQ     220$            ;BRANCH IF HCE IS OK
                                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000          MOVB    #36,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                                220$:
                                                ;REPORT ERROR IF FER IS SET AND HEADER ERRORS ARE NOT ENABLED
                                                        MOV     #-1,-(SP)       ;ASSUME FER IS ENABLED
                                        000536          BIT     #HCE,500$       ;ARE HEADER FRRORS ENABLED ??
                                                        BNE     225$            ;YES !!
                                                        BIC     #FER,(SP)       ;DISABLE FER
                                        001142  225$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;RESET FER IF ENABLED
                                                        BEQ     230$            ;BRANCH IF FER OK
                                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000          MOVB    #37,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                                230$:

                                                ;REPORT ERROR IF BSE IS SET AND HEADER ERRORS ARE NOT ENABLED
                                                        MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                        050536          BIT     #HCE,500$       ;ARE THEY ENABLED ??
                                                        BNE     235$            ;YES .!
                                                        BIC     #BSE,(SP)       ;DISABLE BSE
                                        001142  235$:   MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;CLEAR BSE IF ENABLED
                                                        BEQ     240$            ;BRANCH IF BSE OK
                                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000          MOVB    #354,@(SP)      ;WRITE ERROR NUMBER
                                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                                240$:

                                                ;BIT 06 O, THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR DATA
                                                ;FIELD ERRORS, I.E.,
                                                ;       RMCS2 - MDPE
                                                ;       RMER1 - DCK,ECH
                                                ;NOTE:
                                                ;       ECH CANNOT SET UNLESS IT IS ENABLED AND ECI IS RESET AND
                                                ;DCK IS SET.

                                                ;REPORT ERROR IF MDPE IS SET AND IS NOT ENABLED
                                                        MOV     #-1,-(SP)       ;ASSUME ENABLED
                                        050536          BIT     #ECH,500$       ;ARE DATA FIELD ERRORS ENABLED ??
                                                        BNE     245$            ;YES ..
                                                        BIC     #MDPE,(SP)      ;DISBALE MDPE
                                        001142  245$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;CLEAR MDPE IF ENABLED
```

```
047250  001412                          BEQ     250$            ;BRANCH IF MDPE OK
047252  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
047256  112776  000027  000000          MOVB    #27,@(SP)       ;WRITE ERROR NUMBER IN CALL
047264  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
047270  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
047272  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
047276                          250$:
                                ;REPORT ERROR IF DCK IS SET AND DATA FIELD ERRORS ARE NOT ENABLED
047276  012746  177777                  MOV     #-1,-(SP)       ;ASSUME ENABLED
047302  032737  000100  050536          BIT     #ECH,500$       ;ARE THEY ENABLED ??
047310  001002                          BNE     255$            ;YES !!
047312  042716  100000                  BIC     #DCK,(SP)       ;DISABLE DCK
047316  013737  001352  001142  255$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
047324  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR DCK IF ENABLED
047330  001412                          BEQ     260$            ;BRANCH IF DCK IS OK
047332  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
047336  112776  000030  000000          MOVB    #30,@(SP)       ;WRITE ERROR NUMBER IN CALL
047344  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
047350  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
047352  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
047356                          260$:
                                ;REPORT ERROR IF ECH IS SET AND,
                                ;       DATA FIELD ERRORS ARE NOT ENABLED, OR
                                ;       ECI IS SET, OR
                                ;       DCK IS NOT SET.
047356  012746  177777                  MOV     #-1,-(SP)       ;ASSUME ENABLED
047362  032737  000100  050536          BIT     #ECH,500$       ;ARE ERRORS ENABLED ??
047370  001410                          BEQ     265$            ;NO !!
047372  032737  004000  001370          BIT     #ECI,RMOFI      ;IS ECI SET ??
047400  001004                          BNE     265$            ;YES !!
047402  032737  100000  001352          BIT     #DCK,RMER1I     ;IS DCK ALSO SET ??
047410  001002                          BNE     270$            ;YES !!
047412  042716  000100          265$:   BIC     #ECH,(SP)       ;DISABLE ECH
047416  013737  001352  001142  270$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
047424  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR ECH IF ENABLED
047430  001412                          BEQ     275$            ;BRANCH IF ECH IS OK
047432  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
047436  112776  000031  000000          MOVB    #31,@(SP)       ;WRITE ERROR NUMBER IN CALL
047444  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
047450  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
047452  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
047456                          275$:
```

```
                                        ;;*****************************************************************
                                        ;*PERFORM THE REMAINING ERROR CHECKS ONLY FOR DATA TRANSFER COMMANDS
                                        ;;*****************************************************************

047436  022737  000030  050544            CMP     #SEARCH,515$    ;WAS DATA TRANSFERRED ?
047464  103402                            BLO     280$            ;BR IF YES
047466  000137  050510                    JMP     355$            ;NO - EXIT

                                    ;REPORT ERROR IF RMWC NOT ZERO AND TRE IS ZERO
047472  013737  001340  001142    280$:   MOV     RMWCI,$BDDAT    ;WORD COUNT ZERO??
047500  001421                            BEQ     285$            ;YES
047502  032737  040000  001336            BIT     #TRE,RMCS1I     ;TRANSFER ERROR DETECTED??
047510  001015                            BNE     285$            ;YES.!
047512  062716  000004                    ADD     #4,(SP)
047516  112776  000015  000000            MOVB    #15,@(SP)       ;ERROR NUMBER
047524  005037  001140                    CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
047530  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
047534  004736                            JSR     PC,@(SP)+       ;REPORT WORD COUNT NOT ZERO
047536  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
047542  000240                            NOP

                                    ;REPORT ERROR IF RMBA IS NOT CORRECT
047544  013737  001340  001140    285$:   MOV     RMWCI,$GDDAT    ;GET WORD COUNT AT END OF TRANSFER AND
047552  163737  001414  001140            SUB     RMWCO,$GDDAT    ;SUBTRACT STARTING WORD COUNT.
047560  006337  001140                    ASL     $GDDAT          ;* 2
047564  063737  001416  001140            ADD     RMBAO,$GDDAT    ;ADD STARTING BUS ADDRESS

047572  032737  000010  001346            BIT     #BAI,RMCS2I     ;WAS BUS ADDRESS INHIBIT (BAI) SET ??
047600  001403                            BEQ     290$            ;NO !!
047602  013737  001416  001140            MOV     RMBAO,$GDDAT    ;ADDRESS SHOULD NOT HAVE CHANGED

047610  023737  001140  001342    290$:   CMP     $GDDAT,RMBAI    ;BUS ADDRESS OK??
047616  001416                            BEQ     295$            ;YES!!
047620  013737  001342  001142            MOV     RMBAI,$BDDAT    ;BAD DATA FOR TYPEOUT
047626  062716  000004                    ADD     #4,(SP)
047632  112776  000016  000000            MOVB    #16,@(SP)       ;ERROR NUMBER
047640  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
047644  004736                            JSR     PC,@(SP)+       ;REPORT UNEXPECTED ADDRESS
047646  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
047652  000240                            NOP

                                    ;COMPUTE NUMBER OF SECTORS TRANSFERRED FROM WORD COUNT
047654  005046                    295$:   CLR     -(SP)           ;NUMBER OF SECTORS TRANSFERRED
047656  013746  001340                    MOV     RMWCI,-(SP)     ;GET WORD COUNT AT END OF TRANSFER AND
047662  163716  001414                    SUB     RMWCO,(SP)      ;SUBTRACT STARTING WORD COUNT.

047666  012746  000400                    MOV     #256.,-(SP)     ;ASSUME 256. WORDS PER SECTOR
047672  032737  000002  001412            BIT     #BIT1,RMCS1O    ;HEADER & DATA COMMAND ??
047700  001402                            BEQ     300$            ;NO !!
047702  062716  000002                    ADD     #2,(SP)         ;CHANGE TO 258. WORDS PER SECTOR

047706  005266  000004            300$:   INC     4(SP)           ;INCREMENT SECTOR COUNT
047712  161666  000002                    SUB     (SP),2(SP)      ;SUBTRACT ONE SECTOR'S WORTH
047716  003373                            BGT     300$            ;CONTINUE IF NOT DONE
047720  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK

                                    ;COMPUTE EXPECTED SECTOR, TRACK AND CYLINDER ADDRESS FROM
```

```
                                        ;NUMBER OF SECTORS
047722  013737  001446  050536        MOV     RMDCO,500$      ;STORE ORIGINAL CYLINDER
047730  013737  001420  050540        MOV     RMDAO,505$      ;STORE ORIGINAL TRACK
047736  013737  001420  050542        MOV     RMDAO,510$      ;STORE ORIGINAL SECTOR
047744  013737  001334  050546        MOV     LSTRK,520$      ;STORE LAST TRACK,
047752  000337  050546                SWAB    520$            ;GET TRACK ADDRESS TO LO BYTE AND
047756  005237  050546                INC     520$            ;INCREMENT TO GET TOTL # OF TRACKS.

047762  042737  000377  050540        BIC     #^C<TADMSK>,505$        ;SAVE TRACK ADDRESS BITS AND
047770  000337  050540                SWAB    505$            ;SWAP TRACK ADDRESS TO LOW BYTE.
047774  042737  177400  050542        BIC     #^C<SADMSK>,510$        ;SAVE SECTOR ADDRESS BITS
050002  062637  050542                ADD     (SP)+,510$

050006  023727  050542  000040  310$: CMP     510$,#32.       ;SECTOR OVEFLOWED??
050014  103406                        BLO     315$            ;NO..
050016  005237  050540                INC     505$            ;INCREMENT TRACK
050022  162737  000040  050542        SUB     #32.,510$       ;ADJUST SECTOR
050030  000766                        BR      310$            ;TRY AGAIN

050032  023737  050540  050546  315$: CMP     505$,520$       ;TRACK OVERFLOWED??
050040  103407                        BLO     320$            ;NO.!
050042  005237  050536                INC     500$            ;INCREMENT CYLINDER
050046  163737  050546  050540        SUB     520$,505$       ;ADJUST TRACK
050054  000766                        BR      315$            ;TRY AGAIN
050056  000240                        NOP

                                        ;REPORT ERROR IF "LBT" IS NOT CORRECT
050060                                  320$:
050060  005037  001140                CLR     $GDDAT          ;SET GOOD DATA FOR LBT - 0
050064  023727  050536  001466        CMP     500$,#822.      ;SHOULD LBT BE SET??
050072  101407                        BLOS    325$            ;NO!!
050074  032737  002000  001352        BIT     #IAE,RMER1I     ;WAS IAE SET ??
050102  001003                        BNE     325$            ;YES - LBT SHOULD NOT BE SET
050104  012737  002000  001140        MOV     #LBT,$GDDAT     ;SET GOOD DATA FOR LBT   1
050112  013737  001350  001142  325$: MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
050120  042737  175777  001142        BIC     #^CLBT,$BDDAT
050126  023737  001140  001142        CMP     $GDDAT,$BDDAT   ;IS LBT CORRECT??
050134  001413                        BEQ     330$            ;YES!!
050136  062716  000004                ADD     #4,(SP)
050142  112776  000017  000000        MOVB    #17,@(SP)       ;ERROR NUMBER
050150  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050154  004736                        JSR     PC,@(SP)+       ;REPORT LBT IS WRONG
050156  162716  000010                SUB     #10,(SP)        ;RESTORE (SP)
050162  000240                        NOP

                                        ;REPORT ERROR IF "AOE" IS INCORRECT
050164  005037  001140                330$: CLR     $GDDAT          ;SET FOR AOE = 0
050170  032737  002000  001352        BIT     #IAE,RMER1I     ;WAS "IAE" DETECTED??
050176  001031                        BNE     340$            ;YES-"AOE" SHOULD BE ZERO
050200  023727  050536  001466        CMP     500$,#822.      ;SHOULD AOE BE SET??
050206  101425                        BLOS    340$            ;NO!!
050210  005737  050540                TST     505$            ;MAYBE
050214  001012                        BNE     335$            ;YES
050216  005737  050542                TST     510$
050222  001007                        BNE     335$            ;YES !!
050224  032737  000010  050544        BIT     #F2,515$        ;WAS THIS READ OR WRITE CHECK ??
050232  001413                        BEQ     340$            ;NO ..
```

```
                                                        ;WAS ALL DATA TRANSFERRED ??
                                                        ;YES !.
                                                        ;SET FOR AOE = 1
                                                        ;CLEAR EXPECTED TRACK
                                                        ;EXPECT SECTOR = 1
                                                        ;BAD DATA FOR TYPEOUT
                                                        ;IS AOE CORRECTY??
                                                        ;YES!!
                                                        ;ERROR NUMBER
                                                        ;MOVE SP TO RETURN FOR ERROR
                                                        ;REPORT AOE IS WRONG
                                                        ;RESTORE (SP)

                                ;REPORT ERROR IF RMDA IS NOT CORRECT
                                                        ;WAS THERE AN IAE ERROR ??
                                                        ;YES - DONT CHECK RMDA,RMDC
                                                        ;SETUP EXPECTED DISK ADDRESS
                                                        ;SETUP RECEIVED DISK ADDRESS
                                                        ;COMPARE EXPECTED & RECEIVED
                                                        ;BRANCH IF EQUAL
                                                        ;ERROR NUMBER
                                                        ;MOVE SP TO RETURN FOR ERROR
                                                        ;REPORT BAD DISK ADDRESS
                                                        ;RESTORE (SP)

                                ;REPORT ERROR IF RMDC IS INCORRECT
                                                        ;SETUP EXPECTED CYLINDER
                                                        ;SETUP RECEIVED CYLINDER
                                                        ;COMPARE CYLINDERS
                                                        ;BRANCH IF EQUAL
                                                        ;ERROR NUMBER
                                                        ;MOVE SP TO RETURN FOR ERROR
                                                        ;REPORT BAD CYLINDER
                                                        ;RESTORE (SP)

                                                        ;MOVE (SP) TO ERROR CALL
                                                        ;WAS ERROR FOUND??
                                                        ;MOVE (SP) TO ERROR RETURN
                                                        ;MOVE (SP) TO NO ERROR RETURN

                                                        ;CYLINDER
                                                        ;TRACK
                                                        ;SECTOR
                                                        ;FUNCTION CODE
```

```
        ;? 050546  000000                520$:   .WORD   0               ;TOTAL # OF TRACKS = LAST TRACK +1
```

```
                                .SBTTL  COMPOSITE ERROR CHECK SUBROUTINE
                                
                                ;THIS SUBROUTINE CHECKS THE STORED CONTENTS OF RMER1 AND
                                ;RMER2 AFTER MASKING EACH REGISTER WORD WITH THE USER'S STATUS
                                ;MASKS AND REPORTS AN ERROR IF ANY BITS ARE LEFT ON AFTER
                                ;THE MASKS ARE APPLIED.
                                
                                ;CALL:
                                ;(1)    JSR     PC,CMPERRSTS
                                ;       .WORD                   MASK FOR ERROR REGISTER 1
                                ;       .WORD                   MASK FOR ERROR REGISTER 2
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS
                                
                                ;NOTE:  BITS TO BE MASKED SHOULD BE ONE; BITS TO BE TESTED SHOULD
                                ;BE ZERO
                                
050550                          CMPERRSTS:
                                
                                ;MASK AND STORE THE CONTENTS OF RMER1  AND RMER2
050550  013737  001352  001176          MOV     RMER1I,$TMP1    ;STORE RMER1 AT TEMP STORAGE
050556  047637  000000  001176          BIC     @(SP),$TMP1     ;MASK   RMER1
050564  062716  000002                  ADD     #2,(SP)         ;MOVE SP TO NEXT MASK
050570  013737  001400  001200          MOV     RMER2I,$TMP2    ;STORE RMER2 AT TEMP STORAGE
050576  047637  000000  001200          BIC     @(SP),$TMP2     ;MASK RMER2
                                
                                ;CLEAR USER'S ERROR CALL
050604  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USER'S ERROR CALL
050610  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
050614  162716  000004                  SUB     #4,(SP)         ;LEAVE SP AT NO ERROR RETURN
                                
                                ;SEE IF THERE WERE ANY ERRORS IN RMER1, I.E., $TMP1
050620  005737  001176                  TST     $TMP1           ;ANY ERRORS TO REPORT??
050624  001420                          BEQ     5$              ;NO .!
050626  013737  001176  001142          MOV     $TMP1,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
050634  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
050640  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
050644  112776  000066  000000          MOVB    #66,@(SP)       ;CORRECTABLE DATA CHECK ERROR #
050652  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050656  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
050660  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
050664  000240                          NOP
050666                          5$:
                                
                                ;SEE IF THERE ARE ANY ERRORS TO REPORT IN RMER2 ($TMP2)
050666  005737  001200                  TST     $TMP2           ;ANY ERRORS IN RMER2?
050672  001420                          BEQ     10$             ;NO!!
                                
050674  013737  001200  001142          MOV     $TMP2,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
050702  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
050706  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
050712  112776  000067  000000          MOVB    #67,@(SP)       ;WRITE ERROR NUMBER IN USER'S AC
```

```
                        ;
000720 162716 000002            SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
000724 004736                   JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
000726 162716 000010            SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
000732 000240                   NOP                     ;
000734                  10$:                            ;
                                                        ;
                        ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
000734 062716 000004            ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
000740 105776 000000            TSTB    @(SP)           ;WAS THERE AN ERROR CALLED??
000744 001403                   BEQ     20$             ;NO!!
000746 062716 000004            ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
000752 000402                   BR      30$
000754 162716 000004    20$:    SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
000760 000207           30$:    RTS     PC              ;RETURN TO USER
```

```
                                        .SBTTL  DEVICE SELECT SUBROUTINE
                                        ;
                                        ; THIS SUBROUTINE SELECTS THE DEVICE, GETTING THE DEVICE NUMBER FROM THE
                                        ; TEST QUEUE.
                                        ;
                                        ;CALL:
                                        ;(1)    JSR     PC,DEVSEL
                                        ;(2)    BR      ??              RETURN IF NO ERROR
                                        ;(3)    NOP                     RETURN IF ERROR
                                        ;(4)    ERROR                   ERROR DEFINED BY SUBROUTINE
                                        ;
050762                                  DEVSEL:
                                        ;
                                        ;CLEAR USER'S ERROR CALL
050762  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
050766  105076  000000                          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF CALL
050772  162716  000004                          SUB     #4,(SP)         ;MOVE SP BACK
                                        ;
                                        ;SAVE USER'S INFORMATION AND SETUP REGISTERS
050776  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
051002  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
051006  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
051010  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
051012  012737  051132  000004          MOV     #20$,ERRVEC     ;SETUP FOR BUS TIMEOUT
051020  012737  000300  000006          MOV     #PR6,ERRVEC+2
051026  013700  001276                  MOV     $BASE,R0        ;R0 - UNIBUS ADDRESS
051032  013701  001466                  MOV     TSTQUE,R1       ;R1 POINTS TO DEVICE NUMBER
                                        ;
                                        ;SELECT DEVICE AND VERIFY THAT DEVICE IS AVAILABLE
051036  111160  000010                  MOVB    (R1),RMCS2(R0)  ;WRITE UNIT SELECT BITS
051042  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
051050  016037  000010  001174          MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
                                        ;
051056  032737  010000  001174          BIT     #NED,$TMP0      ;IS DEVICE NONEXISTENT ?
051064  001407                          BEQ     10$             ;NO..
051066  062766  000004  000010          ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
051074  112776  000111  000010          MOVB    #111,@10(SP)    ;WRITE ERROR NUMBER
051102  000422                          BR      30$
                                        ;
051104  032737  004000  001176  10$:    BIT     #DVA,$TMP1      ;IS DEVICE AVAILABLE ?
051112  001021                          BNE     35$             ;YES.!
051114  062766  000004  000010          ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
051122  112776  000112  000010          MOVB    #112,@10(SP)    ;WRITE ERROR NUMBER
051130  000407                          BR      30$
                                        ;
                                        ;HANDLE BUS TIMEOUT
051132  022626                  20$:    CMP     (SP)+,(SP)+     ;ADJUST SP
051134  062766  000004  000010          ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CAL
051142  112776  000113  000010          MOVB    #113,@10(SP)    ;WRITE BUS TIMEOUT ERROR NUMBER
051150  162766  000002  000010  30$:    SUB     #2,10(SP)       ;ADJUST RETURN TO 'NOP' PRECEDING
                                                                ;THE ERROR CALL
                                        ;
                                        ;RESTORE USERS DATA AND RETURN TO ADDRESS ON STACK
051156                                  35$:
051156  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
051160  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
051162  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
```

```
051166  012637  000004                 MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
051172  000207                         RTS     PC              ;EXIT
```

```
                                        .SBTTL  SEEK STATUS CHECK SUBROUTINE
                                        ;THIS SUBROUTINE VERIFIES THE RESULTS OF SEEK TESTS USING STATUS
                                        ;STORED IN THE GET BUFFER AND TEST PARAMETERS STORED IN THE PUT BUFFER.
                                        ;
                                        ;THE SUBROUTINE RETURNS TO THE CALLING ROUTINE IF AN ERROR IS DETECTED
                                        ;AFTER HAVING LOADED THE APPROPRIATE ERROR NUMBER IN THE "ERROR" TRAP
                                        ;OF THE CALLING ROUTINE.  SEEK STATUS IS CHECKED AS FOLLOWS:
                                        ;
                                        ;CALL:
                                        ;(1)    JSR     PC,SEKSTS
                                        ;       BR      ???             RETURN HERE IF NO ERROR
                                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                        ;       ???                     RETURN HERE IF NO MORE ERRORS
                                        ;
051174                                  SEKSTS:

                                        ;CLEAR USERS' ERROR CALL
                                                NOP
051174  000240                                 ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
051176  062716  000004                         CLRB    @(SP)           ;CLEAR ERROR NUMBER
051202  105076  000000                         SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
051206  162716  000004                         CLR     300$            ;CLEAR ERROR FLAGS
051212  005037  052432

                                        ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN WRITING
                                        ;LOCAL REGISTERS, I.E., "PAR"    1 AND "DPE" - 0
051216  032737  000010  001352                 BIT     #PAR,RMER1I     ;WAS PARITY ERROR DETECTED??
051224  001424                                 BEQ     1$              ;NO!!
051226  032737  000010  001400                 BIT     #DPE,RMER2I     ;WAS IT DUE TO CONTROL BUS??
051234  001020                                 BNE     1$              ;NOT SURE!!

                                        ;REPORT REGISTER PARITY ERROR VIA USER'S ERROR CALL
051236  005037  001140                         CLR     $GDDAT          ;EXPECTED STATUS
051242  013737  001352  001142                 MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
051250  062716  000004                         ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
051254  112776  000050  000000                 MOVB    #50,@(SP)       ;ERROR #50
051262  162716  000002                         SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051266  004736                                 JSR     PC,@(SP)+
051270  162716  000010                         SUB     #10,(SP)        ;RESTORE STACK
051274  000437                                 BR      3$              ;IAE SHOULD  BE ZERO

                                        ;DETERMINE THE VALUE OF "IAE" STATUS BASED ON TRACK, SECTOR AND CYLINDER
                                        ;ALSO, SET "SKI" IF CYLINDER ADDRESS IS TOO LARGE.
051276  012737  002000  001140  1$:     MOV     #IAE,$GDDAT     ;SETUP FOR IAE = 1
051304  052737  040000  052432         BIS     #SKI,300$       ;SETUP FOR SKI = 1
051312  023727  001446  001466         CMP     RMDCO,#822.     ;GREATER THAN LAST CYLINDER ?
051320  101025                         BHI     3$              ;YES - CYLINDER IS INVALID
051322  042737  040000  052432         BIC     #SKI,300$       ;CLEAR SKI ERROR FLAG

051330  123737  001421  001335         CMPB    RMDAO+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
051336  101016                         BHI     3$              ;YES - TRACK IS INVALID

051340  123727  001420  000035         CMPB    RMDAO,#29.      ;SECTOR > 29. ?
051346  101410                         BLOS    2$              ;BR IF NO
```

```
                58 051350  032737  010000  001444          BIT     #FMT16,RMDFO    ;18 BIT FORMAT ?
                59 051356  001406                          BEQ     3$              ;YES - SECTOR IS INVALID FOR 18 BIT MODE
                60 051360  123727  001420  000037          CMPB    RMDAO,#31.      ;SECTOR > 31. ?
                61 051366  101002                          BHI     3$              ;YES - SECTOR IS INVALID
                62
                63 051370  005037  001140          2$:     CLR     $GDDAT          ;"IAE" SHOULD = 0
                64
                65                                  ;COMPARE EXPECTED AND RECIEVED "IAE" STATUS
                66 051374  013737  001352  001142  3$:     MOV     RMFR1I,$BDDAT   ;IS IAE OK??
                67 051402  042737  175777  001142          BIC     #^CIAE,$BDDAT   ;SAVE IAE BIT FOR COMPARE
                68 051410  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;CORRECT "IAE" STATUS ?
                69 051416  001004                          BNE     35$             ;BR IF NO
                70 051420  042737  040000  052432          BIC     #SKI,300$       ;CLEAR SKI FLAG
                71 051426  000413                          BR      5$              ;GO CHECK NEXT ERROR
                72 051430                          35$:
                73                                  ;REPORT INCORRECT "IAE" STATUS VIA USER'S ERROR CALL
                74 051430  062716  000004                  ADD     #4,(SP)
                75 051434  112776  000051  000000          MOVB    #51,@(SP)       ;ERROR 51
                76 051442  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                77 051446  004736                          JSR     PC,@(SP)+       ;REPORT INCORRECT IAE
                78 051450  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP)
                79 051454  000240                          NOP
                80 051456                          5$:
                81
                82                                  ;REPORT ANY IVC ERROR AS
                83                                  ;       IVC ERROR WITH VOLUME VALID ZERO
                84                                  ;       ERRONEOUS IVC ERROR, VOLUME VALID IS SET
                85 051456  032737  010000  001400          BIT     #IVC,RMER2I     ;IVC ERROR??
                86 051464  001427                          BEQ     52$             ;NO!!
                87 051466  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
                88 051472  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                89 051500  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                90 051504  112776  000060  000000          MOVB    #60,@(SP)       ;ERROR 60 IF VV - 0
                91 051512  032737  000100  001350          BIT     #VV,RMDSI
                92 051520  001403                          BEQ     51$
                93 051522  112776  000061  000000          MOVB    #61,@(SP)       ;ERROR 61 IF VV  - 1
                94 051530  162716  000002          51$:    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                95 051534  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                96 051536  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
                97 051542  000240                          NOP
                98
                99 051544  013737  001400  001142  52$:    MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
               100 051552  042737  137777  001142          BIC     #^CSKI,$BDDAT   ;CLEAR DONT CARES
               101 051560  013737  052432  001140          MOV     300$,$GDDAT     ;GET EXPECTED SKI STATUS
               102 051566  042737  137777  001140          BIC     #^CSKI,$GDDAT   ;CLEAR DONT CARES
               103 051574  001417                          BEQ     53$             ;BRANCH IF 0 EXPECTED
               104
               105                                  ;REPORT ERROR IF SKI IS NOT SET (IAE WAS NOT DETECTED)
               106 051576  032737  040000  001142          BIT     #SKI,$BDDAT     ;WAS SKI DETECTED ??
               107 051604  001032                          BNE     54$             ;YES !!
               108 051606  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
               109 051612  112776  000267  000000          MOVB    #267,@(SP)      ;WRITE ERROR NUMBER
               110 051620  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
               111 051624  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
               112 051626  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
               113 051632  000443                          BR      6$              ;GO TO NEXT ERROR CHECK
               114 051634                          53$:
```

```
                                        ;REPORT ERROR IF SKI IS SET
051634  032737  040000  001142          BIT     #SKI,$BDDAT     ;IS SKI SET ??
051642  001413                          BEQ     54$             ;NO - SKI IS OK
051644  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
051650  112776  000054  000000          MOVB    #54,@(SP)       ;LOAD ERROR NUMBER
051656  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051662  004736                          JSR     PC,@(SP)+       ;REPORT SEEK ERROR
051664  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
051670  000240                          NOP

                                        ;REPORT ANY DEVICE CHECK
051672  032737  000200  001400  54$:    BIT     #DVC,RMER2I     ;WAS THERE DVC DURING SEEK??
051700  001420                          BEQ     6$              ;NO!!
051702  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
051706  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
051714  062716  000004                  ADD     #4,(SP)
051720  112776  000055  000000          MOVB    #55,@(SP)       ;ERROR  #55
051726  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051732  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
051734  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
051740  000240                          NOP

                                        ;REPORT ANY ''OPI'' ERROR AS OPI WITH MOL    0, OR  OPI
                                        ;BECAUSE ON CYLINDER LATCH DIDN'T RESET
051742  032737  020000  001352  6$:     BIT     #OPI,RMER1I     ;''OPI'' ERROR??
051750  001427                          BEQ     8$              ;NO!!
051752  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
051756  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
051764  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
051770  112776  000052  000000          MOVB    #52,@(SP)       ;LOAD ERROR NUMBER
051776  032737  010000  001350          BIT     #MOL,RMDSI      ;WAS MEDIUM ON LINE??
052004  001403                          BEQ     7$              ;NO!!
052006  112776  000053  000000          MOVB    #53,@(SP)       ;YES - CHANGE ERROR NUMBER
052014  162716  000002          7$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052020  004736                          JSR     PC,@(SP)+       ;REPORT ''OPI'' ERROR
052022  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
052026  000240                          NOP

                                        ;SEE IF ''PIP'' = 0, AND ''ATA'', ''MOL'' AND ''VV'' = 1
052030  013746  001350          8$:     MOV     RMDSI,-(SP)
052034  042716  047677                  BIC     #^C<ATA!PIP.MOL!VV>,(SP)
052040  022726  110100                  CMP     #ATA!MOL!VV,(SP)+
052044  001002                          BNE     9$              ;ERROR IN RMDS
052046  000137  052402                  JMP     14$             ;RMDS IS OK

                                        ;REPORT ERROR IF MOL = 0 AND OPI = 0
052052  032737  010000  001350  9$:     BIT     #MOL,RMDSI      ;IS MOL RESET??
052060  001030                          BNE     10$             ;NO - MOL IS SET
052062  032737  020000  001352          BIT     #OPI,RMER1I     ;WAS OPI SET
052070  001024                          BNE     10$             ;YES - DONT REPORT ERROR
052072  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
052100  052737  010000  001140          BIS     #MOL,$GDDAT
052106  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
052114  062716  000004                  ADD     #4,(SP)
052120  112776  000062  000000          MOVB    #62,@(SP)
052126  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
```

;** STATUS CHECK SUBROUTINE

```
                                                    JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                                    SUB     #10.,(SP)
                                                    NOP

                                        ;REPORT AN ERROR IF "PIP" IS STIL SET AND SKI NOT SET
                                        10$:        BIT     #PIP,RMDSI      ;IS "PIP" STILL SET??
                                                    BEQ     11$             ;NO!!
                                                    BIT     #SKI,RMER2I     ;WAS "SKI"SET??
                                                    BNE     11$             ;YES-DONT REPORT PIP
                                                    MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                    BIC     #PIP,$BDDAT
                                                    MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                    ADD     #4,(SP)         ;MOVE (SP) TO ERROR
                                                    MOVB    #56,@(SP)       ;LOAD ERROR NUMBER
                                                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                    JSR     PC,@(SP)+       ;REPORT "PIP" SET AFTER SEEK
                                                    SUB     #10,(SP)                ;RESTORE (SP)
                                                    NOP

                                        ;REPORT AN ERROR IF "ATA" IS NOT SET
                                        11$:        BIT     #ATA,RMDSI      ;WAS "ATA" SET ??
                                                    BNE     13$             ;YES!!
                                                    MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                    BIS     #ATA!MOL!DPR!DRY,$GDDAT
                                                    MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                    ADD     #4,(SP)         ;MOVE (SP) TO ERROR
                                                    MOVB    #57,@(SP)       ;LOAD ERROR NUMBER
                                                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                    JSR     PC,@(SP)+       ;REPORT ATTENTION NOT SET DURING
                                                                            ;SEEK TEST
                                                    SUB     #10,(SP)                ;RESTORE (SP)
                                                    NOP

                                        ;REPORT ERROR IF VOLUME VALID IS RESET AND IVC IS ZERO
                                        13$:        BIT     #VV,RMDSI       ;IS VV = 0 ??
                                                    BNE     14$             ;NO!!
                                                    BIT     #IVC,RMER2I     ;IS IVC ALSO 0 ??
                                                    BNE     14$             ;NO - IVC IS SET
                                                    MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                    BIS     #VV,$GDDAT
                                                    MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                    ADD     #4,(SP)
                                                    MOVB    #64,@(SP)       ;ERROR #64
                                                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                    JSR     PC,@(SP)+
                                                    SUB     #10,(SP)
                                                    NOP
                                        14$:

                                        ;MODIFY THE RETURN ADDRESS IF AN ERROR WAS DETECTED
                                                    NOP
                                                    ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
                                                    TSTB    @(SP)           ;WAS ERROR CALLED??
                                                    BEQ     15$             ;NO!!
                                                    ADD     #4,(SP)         ;MOVE TO ERROR RETURN
                                                    BR      16$
```

```
    229 052424  162716  000004          15$:    SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    230 052430  000207                  16$:    RTS     PC              ;RETURN
    231
    232 052432  000000                  300$:   .WORD   0               ;ERROR FLAGS
```

```
                        .SBTTL  CONTROLLER CLEAR SUBROUTINE

                        ;THIS SUBROUTINE CLEARS THE MASSBUS CONTROLLER, MASSBUS ADAPTERS,
                        ;AND DRIVES, THEN SELECTS THE DRIVE.
                        ;
                        ;CALL:  JSR     PC,CNTCLR
                        ;       BR      ???             RETURN HERE IF NO ERROR FOUND
                        ;       NOP                     RETURN HERE IF ANY ERROR FOUND
                        ;       ERROR                   SUB DEFINES ERROR NUMBER
                        ;       ???

052434                  CNTCLR:
052434  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
052436  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052440  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
052444  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
052450  012737  052510  000004  MOV     #10$,ERRVEC     ;SETUP FOR BUS TIMEOUT
052456  012737  000300  000006  MOV     #PR6,ERRVEC+2
052464  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
052470  012760  000040  000010  MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
052476  013701  001466          MOV     TSTQUE,R1       ;GET DEVICE UNDER TEST
052502  111160  000010          MOVB    (R1),RMCS2(R0)  ;SELECT DEVICE
052506  000412                  BR      20$

052510  022626          10$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
052512  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
052520  112776  000007  000010  MOVB    #7,@10(SP)      ;WRITE THE ERROR NUMBER
052526  162766  000002  000010  SUB     #2,10(SP)       ;ADJUST SP TO RETURN TO ERROR

052534                  20$:
052534  012637  000006          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
052540  012637  000004          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
052544  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
052546  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
052550  000207                  RTS     PC
```

```
                                        .SBTTL  CONTROLLER CLEAR STATUS CHECK SUBROUTINE

                                        ;THIS SUBROUTINE VERIFIES THAT THE SUBSYSTEM IS INITIALIZED BASED ON
                                        ;STATUS STORED IN THE GET BUFFER.  THIS SUBROUTINE SHOULD ONLY BE
                                        ;USED FOLLOWING A CONTROLLER CLEAR OPERATION, I.E., WRITING A 1 IN BIT
                                        ;5 OF RMCS2, BECAUSE THE ERROR MESSAGES ARE BASED ON THAT CONDITION.

                                        ;STATUS PERTINENT TO THE DEVICE IS NOT CHECKED.  IN PARTICULAR, THE
                                        ;FOLLOWING STATUS BITS ARE NOT CHECKED:
                                        ;
                                        ;       ATA,ERR,PIP,MOL,WRL,LBT,PGM,VV,OM,UNS,SKI,DVC
                                        ;

                                        ;CALL:
                                        ;(1)    JSR     PC,CLRSTS
                                        ;       BR      ???             RETURN HERE IF NO ERROR
                                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                        ;       ???                     RETURN HERE IF NO MORE ERRORS

052552                                  CLRSTS:

                                        ;CLEAR USER'S ERROR CALL
052552  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR
052556  105076  000000                          CLRB    @(SP)           ;CLEAR ERROR NUMBER
052562  162716  000004                          SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR

                                        ;REPORT ERROR IF RMCS1 NOT INITIALIZED
052566  013737  001336  001142        4$:     MOV     RMCS1I,$BDDAT   ;VERIFY RMCS1
052574  042737  100000  001142                BIC     #SC,$BDDAT      ;IGNORE SPECIAL CONDITION
052602  012737  004200  001140                MOV     #DVA!RDY,$GDDAT ;EXPECT DVA & RDY
052610  023737  001140  001142                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED, RECEIVED
052616  001413                                BEQ     5$              ;BRANCH IF EQUAL
052620  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052624  112776  000126  000000                MOVB    #126,@(SP)      ;WITE ERROR NUMBER IN CALL
052632  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052636  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052640  162716  000010                        SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
052644  000240                                NOP
                                        ;REPORT ERROR IF RMBA NOT RESET
052646  005037  001140            5$:     CLR     $GDDAT          ;VERIFY RMBA IS ZERO
052652  013737  001342  001142                MOV     RMBAI,$BDDAT
052660  001413                                BEQ     7$              ;BRANCH IF ZERO
052662  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052666  112776  000127  000000                MOVB    #127,@(SP)      ;WITE ERROR NUMBER IN CALL
052674  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052700  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052702  162716  000010                        SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
052706  000240                                NOP
                                        ;REPORT ERROR IF RMCS2 NOT INITIALIZED
052710  013737  001346  001142        7$:     MOV     RMCS2I,$BDDAT   ;VERIFY RMCS2
052716  010146                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052720  005046                                CLR     -(SP)           ;EXPECT IR & UNIT NUMBER
052722  013701  001466                        MOV     TSTQUE,R1       ;R1 = ADDRESS OF TEST QUE
052726  111116                                MOVB    (R1),(SP)
052730  052716  000100                        BIS     #IR,(SP)
052734  012637  001140                        MOV     (SP)+,$GDDAT
```

```
 58 052740  012601                              MOV     (SP)+,R1        ;;POP STACK INTO R1
 59 052742  023737  001140  001142              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 60 052750  001413                              BEQ     9$              ;BRANCH IF EQUAL
 61 052752  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 62 052756  112776  000130  000000              MOVB    #130,@(SP)      ;WITE ERROR NUMBER IN CALL
 63 052764  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 64 052770  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 65 052772  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 66 052776  000240                              NOP
 67                                     ;REPORT ERROR IF RMER1 NOT RESET-IGNORE UNS
 68 053000  005037  001140              9$:     CLR     $GDDAT          ;VERIFY RMER1
 69 053004  013737  001352  001142              MOV     RMER1I,$BDDAT
 70 053012  042737  040000  001142              BIC     #UNS,$BDDAT     ;IGNORE UNSAFE
 71 053020  001413                              BEQ     13$             ;BRANCH IF ZERO
 72 053022  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 73 053026  112776  000131  000000              MOVB    #131,@(SP)      ;WITE ERROR NUMBER IN CALL
 74 053034  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 75 053040  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 76 053042  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 77 053046  000240                              NOP
 78                                     ;REPORT ERROR IF RMMR1 NOT INITIALIZED-IGNORE WC,LS,LST
 79 053050  013737  001362  001142      13$:    MOV     RMMR1I,$BDDAT   ;VERIFY RMMR
 80 053056  042737  000046  001142              BIC     #WC!LS!LST,$BDDAT       ;IGNORE WORD CLOCK, SCT, TRK
 81 053064  012737  000010  001140              MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA BIT
 82 053072  023737  001140  001142              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
 83 053100  001413                              BEQ     17$             ;BRANCH IF 0
 84 053102  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 85 053106  112776  000133  000000              MOVB    #133,@(SP)      ;WITE ERROR NUMBER IN CALL
 86 053114  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 87 053120  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 88 053122  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 89 053126  000240                              NOP
 90                                     ;REPORT AN ERROR IF RMEC2 IS NOT RESET
 91 053130  005037  001140              17$:    CLR     $GDDAT          ;EXPECT ZEROS
 92 053134  013737  001404  001142              MOV     RMEC2I,$BDDAT   ;VERIFY RMEC2-0
 93 053142  001413                              BEQ     19$
 94 053144  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 95 053150  112776  000135  000000              MOVB    #135,@(SP)      ;WITE ERROR NUMBER IN CALL
 96 053156  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 97 053162  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 98 053164  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 99 053170  000240                              NOP
100                                     ;REPORT ERROR IF RMMR2 NOT INITIALIZED-IGNORE RQA,RQB
101 053172  013737  001376  001142      19$.    MOV     RMMR2I,$BDDAT   ;VERIFY RMMR2
102 053200  042737  140000  001142              BIC     #RQA!RQB,$BDDAT
103 053206  012737  011777  001140              MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
104 053214  023737  001140  001142              CMP     $GDDAT,$BDDAT
105 053222  001413                              BEQ     21$
106 053224  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
107 053230  112776  000136  000000              MOVB    #136,@(SP)      ;WITE ERROR NUMBER IN CALL
108 053236  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
109 053242  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
110 053244  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
111 053250  000240                              NOP
112                                     ;REPORT ERROR IF RMER2 NOT RESET-IGNORE SKI,DVC
113 053252  005037  001140              21$:    CLR     $GDDAT          ;EXPECT ALL ZEROS
114 053256  013737  001400  001142              MOV     RMER2I,$BDDAT   ;VERIFY RMER2
```

```
                                                BIC     #SKI!DVC,$BDDAT ;IGNORE DEVICE ERRORS
                                                BEQ     215$            ;BRANCH IF OTHER BITS 0
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #174,@(SP)      ;WITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
                                                NOP
                                        ;REPORT ERROR IF RMDS NOT INITIALIZED
                                        215$:   MOV     RMDSI,$BDDAT    ;TEST DRIVE STATUS REGISTER
                                                BIC     #^C<DRY!DPR>,$BDDAT
                                                MOV     #DPR.DRY,$GDDAT ;EXPECTED DRIVE STATUS
                                                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
                                                BEQ     22$             ;BRANCH IF EQUAL
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #134,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR TO USER
                                                SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
                                                NOP
                                        22$:    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                TSTB    @(SP)           ;WAS AN ERROE DETECTED??
                                                BEQ     23$             ;NO!.
                                                ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
                                                BR      24$
                                        23$:    SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
                                        24$:    NOP
                                                RTS     PC
```

```
                                .SBTTL  PACK ACKNOWLEDGE STATUS CHECK

                                ;THIS SUBROUTINE CHECKS THE RESULTS OF A PACK ACKNOWLEDGE
                                ;COMMAND USING THE STATUS STORED IN THE GET BUFFER.  ERRORS ARE
                                ;REPORTED TO THE USER VIA THE USER'S ERROR CALL.

                                ;CALL:
                                ;(1)    JSR     PC,ACKSTS
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS

053432                          ACKSTS:

                                ;CLEAR USER'S ERROR CALL
053432  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053436  105076  000000                  CLRB    @(SP)           ;CLEAR LOW ORDER BYTE
053442  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK

                                ;REPORT AN ERROR  IF 'VV' IS 0
053446  032737  000100  001350          BIT     #VV,RMDSI       ;IS VOLUME VALID SET??
053454  001024                          BNE     1$              ;YES!!
053456  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053464  052737  000100  001140          BIS     #VV,$GDDAT
053472  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053500  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053504  112776  000155  000000          MOVB    #155,@(SP)      ;WRITE NUMBER IN ERROR CALL
053512  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053516  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR
053520  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
053524  000240                          NOP
053526                          1$:

                                ;REPORT AN ERROR IF 'MOL" IS 0
053526  032737  010000  001350          BIT     #MOL,RMDSI      ;IS MOL SET??
053534  001024                          BNE     2$              ;YES!!
053536  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053544  052737  010000  001140          BIS     #MOL,$GDDAT
053552  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053560  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053564  112776  000041  000000          MOVB    #41,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
053572  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053576  004736                          JSR     PC,@(SP)+       ;REPORT TH ERROR
053600  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO BRANCH
053604  000240                          NOP
053606                          2$:

                                ;SEE IF "UNS","OPI","RMR","ILR", OR "ILF" IS SET
053606  032737  060007  001352          BIT     #UNS.OPI.RMR.ILR.ILF,RMER1I
053614  001570                          BEQ     7$

                                ;REPORT AN ERROR IF "UNS" IS SET
053616  032737  040000  001352          BIT     #UNS,RMER1I     ;WAS UNS SET??
053624  001424                          BEQ     3$              ;NO..
053626  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

```
 58 053634  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 59 053642  042737  040000  001140          BIC     #UNS,$GDDAT
 60 053650  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 61 053654  112776  000042  000000          MOVB    #42,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
 62 053662  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 63 053666  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 64 053670  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 65 053674  000240                          NOP
 66 053676                          3$.
 67
 68                                 ;REPORT ANY OPI ERROR
 69 053676  032737  020000  001352          BIT     #OPI,RMER1I     ;WAS OPI SET ??
 70 053704  001424                          BEQ     4$              ;NO..
 71 053706  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 72 053714  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 73 053722  042737  020000  001140          BIC     #OPI,$GDDAT
 74 053730  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 75 053734  112776  000043  000000          MOVB    #43,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
 76 053742  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 77 053746  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 78 053750  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 79 053754  000240                          NOP
 80 053756                          4$:
 81
 82                                 ;REPORT ANY RMR ERROR
 83 053756  032737  000004  001352          BIT     #RMR,RMER1I     ;WAS RMR SET??
 84 053764  001424                          BEQ     5$              ;NO!.
 85 053766  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 86 053774  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 87 054002  042737  000004  001140          BIC     #RMR,$GDDAT
 88 054010  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 89 054014  112776  000044  000000          MOVB    #44,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
 90 054022  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 91 054026  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 92 054030  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 93 054034  000240                          NOP
 94 054036                          5$:
 95
 96                                 ;REPORT ANY     ILR ERROR
 97 054036  032737  000002  001352          BIT     #ILR,RMER1I     ;WAS ILR SET??
 98 054044  001424                          BEQ     6$              ;NO!!
 99 054046  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
100 054054  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
101 054062  042737  000002  001140          BIC     #ILR,$GDDAT
102 054070  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
103 054074  112776  000045  000000          MOVB    #45,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
104 054102  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
105 054106  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
106 054110  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
107 054114  000240                          NOP
108 054116                          6$:
109
110                                ;REPORT ANY     ILF ERROR
111 054116  032737  000001  001352          BIT     #ILF,RMER1I     ;WAS ILF SET??
112 054124  001424                          BEQ     7$              ;NO..
113 054126  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
114 054134  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
```

```
115 054142  042737  000001  001140          BIC     #ILF.$GDDAT
116 054150  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
117 054154  112776  000046  000000          MOVB    #46,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
118 054162  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
119 054166  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
120 054170  162716  000010                  SUB     #10.,(SP)       ;MOVE SP TO NO ERROR RETURN
121 054174  000240                          NOP
122 054176                          7$:
123
124                                 ;AUGMENT RETURN ADDRESS IF ERROR WAS FOUND
125 054176  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
126 054202  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
127 054206  001403                          BEQ     8$              ;NO!.
128 054210  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
129 054214  000402                          BR      9$
130 054216  162716  000004          8$:     SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
131 054222  000240                  9$:     NOP
132 054224  000207                          RTS     PC
```

```
                                        .SBTTL  RECALIBRATE STATUS CHECK SUBROUTINE
                                
                                        ;THIS SUBROUTINE CHECKS THE RESULTS OF A RECALIBRATE OPERATION
                                        ;USING THE STATUS STORED IN THE GET BUFFER.
                                
                                        ;CALL:
                                        ;
                                        ;(1)    JSR     PC,RCLSTS       ;CALL SUBROUTINE
                                        ;       BR      ???             RETURN HERE IF NO ERROR
                                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                        ;       ???                     RETURN HERE IF NO MORE ERRORS
                                
054226                                  RCLSTS:
                                
                                        ;CLEAR USER'S ERROR NUMBER
054226  062716  000004                          ADD     #4,(SP)
054232  105076  000000                          CLRB    @(SP)           ;CLEAR USER'S ERROR CALL
054236  162716  000004                          SUB     #4,(SP)         ;MOVE SP BACK TO BRANCH
                                
                                
                                        ;SEE IF "PAR" OR "ILF" OR "OPI" OR "IAE" IS SET
054242  032737  022011  001352                  BIT     #OPI.PAR.ILF.IAE,RMER1I
054250  001553                                  BEQ     4$              ;NONE ARE SET - GO TO NEXT CHECK
                                
                                        ;REPORT ANY MASSBUS CONTROL BUS PARITY ERROR, I.E.,
                                        ;"PAR"  1 AND "DPE" - 0
054252  032737  000010  001352                  BIT     #PAR,RMER1I     ;WAS "PAR" SET??
054260  001430                                  BEQ     1$              ;NO.!
054262  032737  000010  001400                  BIT     #DPE,RMER2I     ;WAS "DPE" SET??
054270  001024                                  BNE     1$              ;YES - NOT A REGISTER ERROR
054272  013737  001352  001140                  MOV RMER1I,$GDDAT       ;EXPECTED STATUS
054300  042737  000010  001140                  BIC     #PAR,$GDDAT
054306  013737  001352  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054314  062716  000004                          ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
054320  112776  000050  000000                  MOVB    #50,@(SP)       ;WRITE ERROR NUMBER IN CALL
054326  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054332  004736                                  JSR     PC,@(SP)+       ;GO REPORT ERROR
054334  162716  000010                          SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
054340  000240                                  NOP
054342                                  1$:
                                
                                        ;REPORT ANY "ILF" ERROR
054342  032737  000001  001352                  BIT     #ILF,RMER1I     ;WAS "ILF" SET??
054350  001424                                  BEQ     2$              ;NO!!
054352  013737  001352  001140                  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054360  042737  000001  001140                  BIC     #ILF,$GDDAT
054366  013737  001352  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054374  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054400  112776  000071  000000                  MOVB    #71,@(SP)       ;WRITE ERROR NUMBER IN CALL
054406  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054412  004736                                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
054414  162716  000010                          SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
054420  000240                                  NOP
054422                                  2$:
```

```
                                        ;REPORT ANY "OPI" ERROR AS
                                        ;       . OPI DUE TO "MOL"   0
                                        ;       . OPI BECAUSE ON CYLINDER LATCH DIDN'T RESET
054422  032737  020000  001352            BIT     #OPI,RMER1I     ;WAS OPI SET??
054430  001433                            BEQ     31$             ;NO!!
054432  013737  001352  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054440  042737  020000  001140            BIC     #OPI,$GDDAT
054446  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054454  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054460  112776  000072  000000            MOVB    #72,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
054466  032737  010000  001350            BIT     #MOL,RMDSI      ;WAS "MOL" = 0??
054474  001403                            BEQ     3$              ;YES!!
054476  112776  000073  000000            MOVB    #73,@(SP)       ;NO - CHANGE ERROR NUMBER
054504  162716  000002          3$:       SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054510  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
054512  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
054516  000240                            NOP
054520                          31$:

                                        ;REPORT AN ERROR IF "IAE" IS SET
054520  032737  002000  001352            BIT     #IAE,RMER1I     ;IS "IAE" SET??
054526  001424                            BEQ     4$              ;NO!!
054530  013737  001352  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054536  042737  002000  001140            BIC     #IAE,$GDDAT
054544  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054552  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054556  112776  000070  000000            MOVB    #70,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
054564  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054570  004736                            JSR     PC,@(SP)+       ;REPORT ERROR
054572  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
054576  000240                            NOP
054600                          4$:

                                        ;SEE IF "SKI" OR "IVC" OR "DVC" IS SET
054600  032737  050200  001400            BIT     #SKI!IVC!DVC,RMER2I
054606  001517                            BEQ     8$              ;NONE OF THE BITS ARE SET


                                        ;REPORT ANY "IVC" ERROR AS
                                        ;       .IVC WITH VV = 0
                                        ;       .ERRONEOUS IVC ERROR
054610  032737  010000  001400            BIT     #IVC,RMER2I     ;WAS IVC SET??
054616  001433                            BEQ     6$              ;NO!!
054620  013737  001400  001140            MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
054626  042737  010000  001140            BIC     #IVC,$GDDAT
054634  013737  001400  001142            MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
054642  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054646  112776  000074  000000            MOVB    #74,@(SP)       ;WRITE ERROR NUMBER IN CALL
054654  032737  000100  001350            BIT     #VV,RMDSI       ;WAS VV = 0??
054662  001403                            BEQ     5$              ;YES!!
054664  112776  000075  000000            MOVB    #75,@(SP)       ;NO - CHANGE ERROR NUMBER
054672  162716  000002          5$:       SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054676  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
054700  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
054704  000240                            NOP
054706                          6$:
```

```
                                        ;REPORT A
054706  032737  040000  001400                BI      #SKI,$GDDAT     ;WAS SKI SET??
054714  001424                              BE      8$              ;NO..
054716  013737  001400  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
054724  042737  040000  001140              BIC     #SKI,$GDDAT
054732  013737  001400  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
054740  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054744  112776  000076  000000              MOVB    #76,@(SP)       ;WRITE ERROR NUMBER
054752  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054756  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
054760  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO BRANCH
054764  000240                              NOP
054766                                  7$:

                                        ;REPORT ANY 'DVC' ERROR
054766  032737  000200  001400                BIT     #DVC,RMER2I     ;WAS 'DVC' SET??
054774  001424                              BEQ     8$              ;NO!!
054776  013737  001400  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
055004  042737  000200  001140              BIC     #DVC,$GDDAT
055012  013737  001400  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
055020  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055024  112776  000077  000000              MOVB    #77,@(SP)       ;WRITE ERROR NUMBER
055032  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055036  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055040  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
055044  000240                              NOP
055046                                  8$:

                                        ;SEE IF 'PIP' AND 'OM' ARE 0, AND 'ATA','MOL' AND 'VV' ARE 1
055046  013746  001350                      MOV     RMDSI,-(SP)     ;PUT RMDS ON STACK
055052  042716  047676                      BIC     #^C<PIP!MOL!VV!OM!ATA>,(SP)
055056  022726  110100                      CMP     #ATA!MOL!VV,(SP)+
055062  001002                              BNE     85$
055064  000137  055500                      JMP     13$
055070                                  85$:

                                        ;REPORT AN ERROR IF MOL = 0 AND OPI = 0, I.E , MEDIUM WENT OFF
                                        ;LINE AFTER RECALIBRATE WAS INITIATED
055070  032737  010000  001350                BIT     #MOL,RMDSI      ;DID MOL DROP??
055076  001030                              BNE     9$              ;NO!!
055100  032737  020000  001352              BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
055106  001024                              BNE     9$              ;YES - DON'T REPORT MOL 0
055110  013737  001350  001140              MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
055116  052737  010000  001140              BIS     #MOL,$GDDAT
055124  013737  001350  001142              MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
055132  062716  000004                      ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
055136  112776  000100  000000              MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
055144  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055150  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055152  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
055156  000240                              NOP
055160                                  9$:

                                        ;REPORT AN ERROR IF 'VV' = 0 AND 'IVC' = 0
055160  032737  000100  001350                BIT     #VV,RMDSI       ;DID 'VV' DROP??
055166  001030                              BNE     10$             ;NO!.
055170  032737  010000  001400              BIT     #IVC,RMER2I     ;WAS THERE A IVC ERROR??
```

```
                                        BNE     10$             ;YES - DONT REPORT VV-0
                                        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                        BIS     #VV,$GDDAT
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #101,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+
                                        SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
                                        NOP
                                10$:

                                ;REPORT AN ERROR IF ATA IS NOT SET
                                        BIT     #ATA,RMDSI      ;WAS ATA SET DURING RECALIBRATE??
                                        BNE     11$             ;YES!!
                                        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                        BIS     #ATA,$GDDAT
                                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #102,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)
                                        JSR     PC,@(SP)+
                                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
                                        NOP

                                11$:

                                ;REPORT AN ERROR IF "OM" IS NOT ZERO BECAUSE RECALIBRATE SHOULD
                                ;ALWAYS CLEAR OFFSET MODE
                                        BIT     #OM,RMDSI       ;WAS "OM" RESET??
                                        BEQ     12$             ;YES!!
                                        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                        BIC     #OM,$GDDAT
                                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #103,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
                                        NOP
                                12$:

                                ;REPORT AN ERROR IF "PIP" IS STIL ON, I.E., DRIVE NOT ON
                                ;CYLINDER
                                        BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
                                        BEQ     13$             ;NO!!
                                        BIT     #SKI,RMER2I     ;WAS "SKI" DETECTED??
                                        BNE     13$             ;YES-DONT REPORT "PIP"
                                        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                        BIC     #PIP,$GDDAT
                                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                        ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #104,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+
                                        SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
                                        NOP
```

```
055500                                  13$:

                                        ;SEE IF "ILR" OR "RMR" OR "UNS" IS SET
055500  032737  040006  001352                  BIT     #ILR!RMR!UNS,RMER1I
055506  001514                                  BEQ     16$

                                        ;REPORT AN ERROR IF "ILR" IS SET
055510  032737  000002  001352                  BIT     #ILR,RMER1I     ;WAS ILR SET DURING RECALIBRATE??
055516  001424                                  BEQ     14$             ;NO!!
055520  013737  001352  001140                  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055526  042737  000002  001140                  BIC     #ILR,$GDDAT
055534  013737  001352  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055542  062716  000004                          ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
055546  112776  000105  000000                  MOVB    #105,@(SP)      ;WRITE ERROR NUMBER IN CALL
055554  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055560  004736                                  JSR     PC,@(SP)+
055562  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
055566  000240                                  NOP
055570                                  14$:

                                        ;REPORT AN ERROR IF "RMR" IS SET
055570  032737  000004  001352                  BIT     #RMR,RMER1I     ;WAS RMR SET??
055576  001424                                  BEQ     15$             ;NO!!
055600  013737  001352  001140                  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055606  042737  000004  001140                  BIC     #RMR,$GDDAT
055614  013737  001352  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055622  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055626  112776  000106  000000                  MOVB    #106,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
055634  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055640  004736                                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055642  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
055646  000240                                  NOP
055650                                  15$:

                                        ;REPORT AN ERROR IF "UNS" IS SET AND "DVC" IS 0
055650  032737  040000  001352                  BIT     #UNS,RMER1I     ;WAS UNSAFE ON??
055656  001430                                  BEQ     16$             ;NO:!
055660  032737  000200  001400                  BIT     #DVC,RMER2I     ;WAS THERE A DEVICE CHECK??
055666  001024                                  BNE     16$             ;YES - DON'T REPORT UNSAFE
055670  013737  001352  001140                  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055676  042737  040000  001140                  BIC     #UNS,$GDDAT
055704  013737  001352  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055712  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055716  112776  000107  000000                  MOVB    #107,@(SP)      ;WRITE ERROR NUMBER
055724  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055730  004736                                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055732  162716  000010                          SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
055736  000240                                  NOP
055740                                  16$:

                                        ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
055740  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055744  105776  000000                          TSTB    @(SP)           ;WAS AN ERROR REPORTED??
055750  001403                                  BEQ     17$             ;NO!!
055752  062716  000004                          ADD     #4,(SP)         ;YES - AUGMENT SP RETURN
055756  000402                                  BR      18$
055760  162716  000004                  17$:    SUB     #4,(SP) ;NO ERROR - RETURN SP TO BRANCH
```

```
 286 055764  000240                  18$:    NOP
 287 055766  000207                          RTS     PC              ;STATUS CECK IS COMPLETE
```

```
                                        .SBTTL  DRIVE CLEAR STATUS CHECK SUBROUTINE
                                        ;
                                        ;       BR      ???             RETURN HERE IF NO ERROR
                                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                        ;       ???                     RETURN HERE IF NO MORE ERRORS
                                        ;
055770                                  DRVSTS:
                                        ;
                                        ;CLEAR USER'S ERROR CALL
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
055770  062716  000004                        CLRB    @(SP)           ;CLEAR ERROR CALL
056000  162716  000004                        SUB     #4,(SP)         ;MOVE SP TO USER'S BRANCH
                                        ;REPORT ERROR IF RMCS1 NOT INITIALIZED
056004  013737  001336  001142        4$:     MOV     RMCS1I,$BDDAT   ;CHECK RMCS1
056012  042737  173700  001142                BIC     #^C<DVA!FNCMSK>,$BDDAT  ;CLEAR DONT CARES
056020  012737  004010  001140                MOV     #DVA!DRVCLR,$GDDAT      ;EXPECT DVA
056026  023737  001140  001142                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
056034  001443                                BEQ     6$              ;BRANCH IF EQUAL
056036  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
056042  112776  000141  000000                MOVB    #141,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
056050  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056054  004736                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
056056  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
056062  000240                                NOP
                                        ;REPORT ERROR IF RMDS NOT INITIALIZED
056064  013737  001350  001142        5$:     MOV     RMDSI,$BDDAT    ;CHECK RMDS
056072  042737  021101  001142                BIC     #PGM!OM!VV!PIP,$BDDAT   ;CLEAR DONT CARES
056100  012737  010600  001140                MOV     #MOL!DPR!DRY,$GDDAT     ;EXPECT DRY & DPR
056106  023737  001140  001142                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
056114  001413                                BEQ     6$              ;BRANCH IF EQUAL
056116  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
056122  112776  000142  000000                MOVB    #142,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
056130  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056134  004736                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
056136  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
056142  000240                                NOP
                                        ;REPORT ERROR IF RMER1 NOT INITIALIZED
056144  005037  001140                6$:     CLR     $GDDAT          ;EXPECT 0'S
056150  013737  001352  001142                MOV     RMER1I,$BDDAT   ;CHECK RMER1
056156  001413                                BEQ     8$              ;BRANCH IF EQUAL
056160  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
056164  112776  000143  000000                MOVB    #143,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
056172  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056176  004736                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
056200  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
056204  000240                                NOP
                                        ;REPORT ERROR IF ATA NOT INITIALIZED
056206  013737  001354  001142        8$:     MOV     RMASI,$BDDAT    ;CHECK ATTENTION BIT
056214  010146                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
056216  010246                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
056220  013701  001466                        MOV     TSTQUE,R1
056224  116102  000001                        MOVB    1(R1),R2
056230  042702  177400                        BIC     #^CATNMSK,R2
056234  005102                                COM     R2
056236  040237  001142                        BIC     R2,$BDDAT
```

```
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        TST     $BDDAT          ;IS ATTENTION CLEARED??
                                        BEQ     9$              ;BRANCH IF ATTENTION CLEARED
                                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                        MOVB    #144,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
                                ;REPORT ERROR IF RMMR1 NOT INITIALIZED
                                9$:     MOV     RMMR1I,$BDDAT   ;CHECK RMMR
                                        BIC     #WC!LS!LST,$BDDAT       ;CLEAR DONT CARES
                                        MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA ON
                                        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                        BEQ     11$             ;BRANCH IF ZERO
                                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                        MOVB    #145,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
                                ;REPORT ERROR IF RMMR2 NOT INITIALIZED
                                11$:    MOV     RMMR2I,$BDDAT   ;CHECK RMMR2
                                        BIC     #RQA!RQB,$BDDAT ;CLEAR REQA, REQB
                                        MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
                                        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
                                        BEQ     15$             ;BRANCH IF EQUAL
                                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                        MOVB    #146,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
                                15$:    CLR     $GDDAT          ;EXPECT ZEROS
                                ;REPORT ERROR IF RMEC2 NOT RESET
                                        MOV     RMEC2I,$BDDAT   ;CHECK RMEC2
                                        BEQ     17$             ;BRANCH IF 0
                                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                        MOVB    #150,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
                                ;REPORT ERROR IF RMER2 NOT RESET
                                17$:    MOV     RMER2I,$BDDAT   ;CHECK RMER2
                                        BEQ     18$             ;BRANCH IF NO ERROR
                                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                        MOVB    #147,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
                                18$:

                                19$:

```

```
                                ;AUGMENT RETURN ADDRESS IF ANY ERROR WAS FOUND
056542  062716  000004            ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
056546  105776  000000            TSTB    @(SP)           ;WAS AN ERROR DETECTED??
056552  001403                    BEQ     21$             ;NO.!
056554  062716  000004            ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
056560  000402                    BR      23$
056562  162716  000004    21$:    SUB     #4,(SP) ;MOVE SP BACK TO NO ERROR RETURN
056566  000240    23$:    NOP
05657C  000207                    RTS     PC              ;RETURN TO USER
```

```
                                        .SBTTL  DATA TRANSFER COMMAND STATUS CHECK SUBROUTINE

                                        ;THIS SUBROUTINE VERIFIES THE RESULTS OF ALL DATA TRANSFER COMMANDS
                                        ;USING STATUS STORED IN THE GET BUFFER AND TEST PARAMETERS
                                        ;STORED IN THE PUT BUFFER.  ERRORS ARE REPORTED BY WRITING
                                        ;THE ERROR NUMBER IN THE USERS ERROR CALL.

                                        ;USER'S SUBROUTINE CALL:
                                        ;(1)    JSR     PC,DTASTS
                                        ;(2)    BR      ??              RETURN HERE IF NO DATA ERRORS
                                        ;(3)    NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;(4)    ERROR                   SUB WRITES ERROR NUMBER
                                        ;(5)    JSR     PC,@(SP)+       USER RETURNS FOR MORE CHECKS
                                        ;(6)    ??                      SUB RETURNS HERE AFTER ALL
                                        ;                               ERRORS ARE REPORTED

056572                                  DTASTS:

                                        ;CLEAR USER'S ERROR CALL AND ERROR FLAGS
056572  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
056576  105076  000000                          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF TRAP
056602  162716  000004                          SUB     #4,(SP)         ;RESTORE SP TO NO ERROR
056606  005037  062166                          CLR     500$            ;CLEAR ERROR FLAGS

                                        ;REPORT ANY CONTROL BUS PARITY ERROR WHILE READING REMOTE REGISTERS,
                                        ;I.E., MCPE = 1
056612  032737  020000  001336                  BIT     #MCPE,RMCS1I    ;WAS THERE A PARITY ERROR??
056620  001422                                  BEQ     10$             ;NO!.
056622  013737  001336  001140                  MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
056630  042737  020000  001140                  BIC     #MCPE,$GDDAT
056636  013737  001336  001142                  MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
056644  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056650  112776  000013  000000                  MOVB    #13,@(SP)       ;WRITE ERROR NUMBER
056656  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056662  004736                                  JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056664  000466                                  BR      30$
056666                                  10$:

                                        ;REPORT ANY CONTROL BUS PARITY ERROR WHILE WRITTING REMOTE REGISTERS,
                                        ;I.E., PAR - 1 AND DPE = 0
056666  032737  000010  001352                  BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
056674  001435                                  BEQ     20$             ;NO!!
056676  032737  000010  001400                  BIT     #DPE,RMER2I     ;DATA PARITY ERROR ?
056704  001031                                  BNE     20$             ;YES!!
056706  013737  001352  001140                  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056714  042737  000010  001140                  BIC     #PAR,$GDDAT
056722  013737  001352  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056730  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
056734  112776  000050  000000                  MOVB    #50,@(SP)       ;WRITE ERROR NUMBER
056742  032737  001000  001346                  BIT     #MXF,RMCS2I     ;DID MXF GET SET??
056750  001003                                  BNE     15$             ;YES!!
056752  112776  000274  000000                  MOVB    #274,@(SP)      ;NO - CHANGE ERROR NUMBER
056760  162716  000002                  15$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056764  004736                                  JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056766  000425                                  BR      30$

                                        ;LOOK FOR ANY ERRORS WHICH MAY HAVE OCCURRED DURING COMMAND INITIATION R
```

```
                                        ;MECHANICAL POSITIONING
                                        ;
                                        ;FIRST TEST MXF WHICH WOULD INDICATE COMPOSITE ERROR SET WHEN FUNCTION
                                        ;CODE AND GO BIT WERE LOADED
056770                                  20$:
056770  032737  001000  001346                BIT     #MXF,RMCS2I     ;WAS 'MISSED TRANSFER'' SET??
056776  001425                              BEQ     40$             ;NO..
057000  013737  001346  001140              MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
057006  042737  001000  001140              BIC     #MXF,$GDDAT
057014  013737  001346  001142              MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
057022  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057026  112776  000275  000000              MOVB    #275,@(SP)      ;WRITE ERROR NUMBER
057034  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057040  004736                              JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057042                                  30$:
                                        ;
                                        ;RESTORE SP TO NO ERROR RETURN AND BYPASS FURHTER STATUS CHECKING
057042  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO NO ERROR
057046  000137  062140                      JMP     380$            ;SKIP TO END OF SUB
                                        ;
057052                                  40$:
                                        ;
                                        ;REPORT AN ERROR IF ''OPI'' ERROR OCCURRED DUE TO ''MOL'' = 0, OR IF ''OPI''
                                        ;AND ''MOL'' ARE SET, BUT ''VV'' IS RESET, INDICATING AN INTERMITTENT
                                        ;''MOL''
057052  032737  020000  001352              BIT     #OPI,RMER1I     ;IS ''OPI'' SET??
057060  001447                              BEQ     60$             ;NO.!
057062  013737  001352  001140              MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
057070  042737  020000  001140              BIC     #OPI,$GDDAT
057076  013737  001352  001142              MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
057104  032737  010000  001350              BIT     #MOL,RMDSI      ;WAS MEDIUM OFF LINE??
057112  001404                              BEQ     45$             ;YES!!
057114  032737  000100  001350              BIT     #VV,RMDSI       ;WAS ''MOL'' INTERMITTENT??
057122  001013                              BNE     50$             ;NO!!
057124  062716  000004              45$:    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057130  112776  000276  000000              MOVB    #276,@(SP)      ;WRITE ERROR NUMBER IN CALL
057136  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057142  004736                              JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057144  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
057150  000413                              BR      60$
057152                                  50$:
                                        ;
                                        ;REPORT ''OPI'' ERROR, WHICH IS DUE TO ''ON CYLINDER'' NOT DROPPING OR
                                        ;''RUN'' TIMEOUT (20 MS) OR SEARCH TIMEOUT (50 MS)
057152  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057156  112776  000277  000000              MOVB    #277,@(SP)      ;WRITE ERROR NUMBER IN CALL
057164  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057170  004736                              JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057172  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
057176  000240                              NOP
057200                                  60$:
                                        ;
                                        ;LOOK FOR ''IVC'' ERROR DURING COMMAND INITIATION
057200  032737  010000  001400              BIT     #IVC,RMER2I     ;WAS THERE AN ''IVC'' ERROR??
057206  001432                              BEQ     70$             ;NO..
                                        ;REPORT ''IVC'' ERROR DUE TO ''VV'' = 0, OR REPORT ERRONEOUS ''I,'' ERROR
057210  013737  001400  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
```

```
115 057216  042737  010000  001140          BIC     #IVC,$GDDAT
116 057224  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
117 057232  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
118 057236  112776  000300  000000          MOVB    #300,@(SP)      ;WRITE ERROR NUMBER IN CALL
119 057244  032737  000100  001350          BIT     #VV,RMDSI       ;WAS VOLUME VALID??
120 057252  001403                          BEQ     65$             ;NO..
121 057254  112776  000301  000000          MOVB    #301,@(SP)      ;CHANGE ERROR NUMBER
122 057262  162716  000002          65$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
123 057266  004736                          JSR     PC,@(SP)+       ;REPORT ''IVC'' ERROR AND RETURN
124 057270  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
125 057274                          70$:
126
127                                 ;SEE IF ''ILF'' OR 'RMR'' IS SET
128 057274  032737  000007  001352          BIT     #ILR!ILF.RMR, RMER1I
129 057302  001510                          BEQ     100$            ;NO ERRORS DETECTED
130                                 ;REPORT AN ERROR IF ''ILR'' IS SET
131 057304  032737  000002  001352          BIT     #ILR,RMER1I     ,WAS ''ILR'' DETECTED??
132 057312  001424                          BEQ     80$             ;NO.!
133 057314  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
134 057322  042737  000002  001140          BIC     #ILR,$GDDAT
135 057330  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
136 057336  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
137 057342  112776  000302  000000          MOVB    #302,@(SP)      ;WRITE ERROR NUMBER IN CALL
138 057350  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
139 057354  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
140 057356  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
141 057362  000240                          NOP
142 057364                          80$:
143
144                                 ;REPORT AN ERROR IF ''ILF'' IS SET
145 057364  032737  000001  001352          BIT     #ILF,RMER1I     ;WAS ''ILF'' DETECTED??
146 057372  001424                          BEQ     90$             ;NO.!
147 057374  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
148 057402  042737  000001  001140          BIC     #ILF,$GDDAT
149 057410  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
150 057416  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
151 057422  112776  000303  000000          MOVB    #303,@(SP)      ;WRITE ERROR NUMBER IN CALL
152 057430  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
153 057434  004736                          JSR     PC,@(SP)+       ;REPORT ERORR AND RETURN
154 057436  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
155 057442  000240                          NOP
156 057444                          90$:
157
158                                 ;REPORT AN ERROR IF 'RMR'' IS SET
159 057444  032737  000004  001352          BIT     #RMR,RMER1I     ;WAS 'RMR'' DETECTED??
160 057452  001424                          BEQ     100$            ;NO!.
161 057454  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
162 057462  042737  000004  001140          BIC     #RMR,$GDDAT
163 057470  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
164 057476  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
165 057502  112776  000304  000000          MOVB    #304,@(SP)      ;WRITE ERROR NUMBER IN CALL
166 057510  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
167 057514  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
168 057516  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
169 057522  000240                          NOP
170 057524                          100$:
171                                 ;DETERMINE WHETHER OR NOT ''IAE'' SHOULD BE SET AND CHECK FOR ERROR
```

```
                        ;
                        ;
                        ;
057524 012737 002000 001140        MOV     #IAE,$GDDAT     ;SETUP FOR ''IAE''  1
057532 052737 040000 062166        BIS     #SKI,500$       ;SETUP FOR ''SKI'' - 1
057540 023727 001446 001466        CMP     RMDCO,#822.     ;GREATER THAN LAST CYLINDER ?
057546 101025                      BHI     110$            ;YES - CYLINDER IS INVALID
057550 042737 040000 062166        BIC     #SKI,500$       ;RESET SKI FLAG
                        ;
057556 123737 001421 001335        CMPB    RMDAO+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
057564 101016                      BHI     110$            ;YES - TRACK IS INVALID
                        ;
057566 123727 001420 000035        CMPB    RMDAO,#29.      ;IS SECTOR > 29. ?
057574 101410                      BLOS    105$            ;NO
057576 032737 010000 001444        BIT     #FMT16,RMOFO    ;18 BIT FORMAT ?
057604 001406                      BEQ     110$            ;YES - SECTOR IS INVALID FOR 18 BIT MODE
057606 123727 001420 000037        CMPB    RMDAO,#31.      ;IS SECTOR > 31. ?
057614 101002                      BHI     110$            ;YES - SECTOR IS INVALID
057616 005037 001140        105$:  CLR     $GDDAT          ;''IAE'' SHOULD = 0
                        ;
057622 013737 001352 001142 110$:  MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
057630 042737 175777 001142        BIC     #^CIAE,$BDDAT
057636 023737 001140 001142        CMP     $GDDAT,$BDDAT   ;IS ''IAE'' STATUS OK??
057644 001004                      BNE     115$            ;NO!.
057646 042737 040000 062166        BIC     #SKI,500$       ;IAE OK - SKI SHOULD BE 0
057654 000412                      BR      120$
057656 062716 000004        115$:  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057662 112776 000305 000000        MOVB    #305,@(SP)      ;WRITE ERROR NUMBER
057670 162716 000002               SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057674 004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057676 162716 000010               SUB     #10,(SP)        ;MOVE SP TO NO ERROR
057702                      120$:
                        ;
                        ;REPORT AN ERROR IF ''SKI'' IS SET AND ''IAE'' STATUS WAS OK
057702 013737 001400 001142        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
057710 042737 137777 001142        BIC     #^CSKI,$BDDAT
057716 013737 062166 001140        MOV     500$,$GDDAT     ;EXPECTED STATUS
057724 042737 137777 001140        BIC     #^CSKI,$GDDAT
057732 032737 040000 001400        BIT     #SKI,RMER2I     ;WAS ''SKI'' SET??
057740 001417                      BEQ     140$            ;NO!!
057742 032737 040000 062166        BIT     #SKI,500$       ;WAS SKI CAUSED BY IAE - 0??
057750 001032                      BNE     150$            ;YES - DON'T REPORT SKI
057752 062716 000004               ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
057756 112776 000306 000000        MOVB    #306,@(SP)      ;WRITE ERROR NUMBER
057764 162716 000002               SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057770 004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057772 162716 000010               SUB     #10,(SP)        ;MOVE SP TO NO ERROR
057776 000417                      BR      150$
                        ;
060000                      140$:
                        ;
                        ;REPORT AN ERROR IF SKI   0 AND IAE WAS NOT DETECTED
060000 032737 040000 062166        BIT     #SKI,500$       ;SHOULD SKI BE SET??
060006 001413                      BEQ     150$            ;NO!.
060010 062716 000004               ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060014 112776 000307 000000        MOVB    #307,@(SP)      ;WRITE ERROR NUMBER IN CALL
060022 162716 000002               SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060026 004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060030 162716 000010               SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
060034 000240                      NOP
```

```
 229 060036                                  150$:
 230
 231                                         ;LOOK FOR 'LSC' OR 'LBC' OR 'DVC' IN ERROR REGISTER #2
 232 060036  032737  006200  001400                  BIT     #LSC!LBC!DVC,RMER2I
 233 060044  001512                                  BEQ     180$                ;NO ERRORS SET
 234
 235                                         ;REPORT ANY DEVICE FAULT, I.E., 'DVC' = 1
 236 060046  032737  000200  001400                  BIT     #DVC,RMER2I         ;IS 'DVC' = 1??
 237 060054  001424                                  BEQ     160$                ;NO!!
 238 060056  013737  001400  001140                  MOV     RMER2I,$GDDAT       ;EXPECTED STATUS
 239 060064  042737  000200  001140                  BIC     #DVC,$GDDAT
 240 060072  013737  001400  001142                  MOV     RMER2I,$BDDAT       ;RECEIVED STATUS
 241 060100  062716  000004                          ADD     #4,(SP)             ;MOVE SP TO USERS ERROR
 242 060104  112776  000310  000000                  MOVB    #310,@(SP)          ;WRITE ERROR NUMBER IN CALL
 243 060112  162716  000002                          SUB     #2,(SP)             ;MOVE SP TO RETURN IF ERROR
 244 060116  004736                                  JSR     PC,@(SP)+           ;REPORT ERROR AND RETURN
 245 060120  162716  000010                          SUB     #10,(SP)            ;RESTORE SP TO NO ERROR
 246 060124  000240                                  NOP
 247 060126                                  160$:
 248
 249                                         ;REPORT LOSS OF BIT CLOCK, I.E.; 'LBC' = 1, IF 'MOL' = 1
 250 060126  032737  002000  001400                  BIT     #LBC,RMER2I         ;IS LBC SET??
 251 060134  001430                                  BEQ     170$                ;NO!!
 252 060136  032737  010000  001350                  BIT     #MOL,RMDSI          ;WAS LBC ERROR BY MOL = 0
 253 060144  001424                                  BEQ     170$                ;YES!!
 254 060146  013737  001400  001140                  MOV     RMER2I,$GDDAT       ;EXPECTED STATUS
 255 060154  042737  002000  001140                  BIC     #LBC,$GDDAT
 256 060162  013737  001400  001142                  MOV     RMER2I,$BDDAT       ;RECEIVED STATUS
 257 060170  062716  000004                          ADD     #4,(SP)             ;MOVE SP TO USER'S ERROR CALL
 258 060174  112776  000311  000000                  MOVB    #311,@(SP)          ;WRITE ERROR NUMBER IN CALL
 259 060202  162716  000002                          SUB     #2,(SP)             ;MOVE SP TO RETURN IF ERROR
 260 060206  004736                                  JSR     PC,@(SP)+           ;REPORT ERROR AND RETURN
 261 060210  162716  000010                          SUB     #10,(SP)            ;RESTORE SP TO NO ERROR
 262 060214  000240                                  NOP
 263 060216                                  170$:
 264
 265                                         ;REPORT LOS OF SYSTEM CLOCK, I.E., 'LSC' = 1
 266 060216  032737  004000  001400                  BIT     #LSC,RMER2I         ;IS 'LSC' = 1??
 267 060224  001422                                  BEQ     180$                ;NO!!
 268 060226  013737  001400  001140                  MOV     RMER2I,$GDDAT       ;EXPECTED STATUS
 269 060234  042737  004000  001140                  BIC     #LSC,$GDDAT
 270 060242  013737  001400  001142                  MOV     RMER2I,$BDDAT       ;RECEIVED STATUS
 271 060250  062716  000004                          ADD     #4,(SP)             ;MOVE SP TO USER'S ERROR CALL
 272 060254  112776  000312  000000                  MOVB    #312,@(SP)          ;WRITE ERROR NUMBER
 273 060262  004736                                  JSR     PC,@(SP)+           ;REPORT ERROR AND RETURN
 274 060264  162716  000010                          SUB     #10,(SP)            ;RESTORE SP TO NO ERROR
 275 060270  000240                                  NOP
 276 060272                                  180$:
 277
 278                                         ;LOOK FOR 'UNS' OR 'DTE' OR 'WLE' IN ERROR REGISTER #1
 279 060272  032737  054000  001352                  BIT     #UNS!DTE.WLE,RMER1I
 280 060300  001527                                  BEQ     220$                ;NO BITS SET
 281                                         ;REPORT 'UNS' ERROR IF 'DVC' - 0
 282 060302  032737  040000  001352                  BIT     #UNS,RMER1I         ;IS 'UNS' SET??
 283 060310  001427                                  BEQ     190$                ;NO!.
 284 060312  032737  000200  001400                  BIT     #DVC,RMER2I         ;WAS 'UNS' CAUSED BY 'DVC'??
 285 060320  001023                                  BNE     190$                ;YES..
```

```
 286 060322  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 287 060330  042737  040000  001140          BIC     #UNS,$GDDAT
 288 060336  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 289 060344  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 290 060350  112776  000313  000000          MOVB    #313,@(SP)      ;WRITE ERROR NUMBER
 291 060356  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 292 060362  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 293 060364  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
 294 060370                          190$:
 295
 296                                 ;REPORT ANY DRIVE TIMING ERROR, I.E., 'DTE' = 1
 297 060370  032737  010000  001352          BIT     #DTE,RMER1I     ;IS DTE SET??
 298 060376  001423                          BEQ     200$            ;NO!!
 299 060400  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 300 060406  042737  010000  001140          BIC     #DTE,$GDDAT
 301 060414  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 302 060422  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 303 060426  112776  000314  000000          MOVB    #314,@(SP)      ;WRITE ERROR NUMBER IN CALL
 304 060434  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 305 060440  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 306 060442  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 307 060446                          200$:
 308
 309                                 ;REPORT AN ERROR IF WRITE LOCK ERROR IS SET.  SEE IF DRIVE IS NOT
 310                                 ;WRITE PROTECTED, OR IF FUNCTION WAS NOT A WRITE
 311 060446  032737  004000  001352          BIT     #WLE,RMER1I     ;WAS 'WLE' SET??
 312 060454  001441                          BEQ     220$            ;NO!!
 313 060456  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 314 060464  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 315 060472  052737  004000  001140          BIS     #WLE,$GDDAT
 316 060500  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 317 060504  112776  000315  000000          MOVB    #315,@(SP)      ;WRITE ERROR NUMBER IN CALL
 318 060512  032737  004000  001350          BIT     #WRL,RMDSI      ;WAS DRIVE WRITE PROTECTED??
 319 060520  001404                          BEQ     205$            ;NO!!
 320 060522  032737  000010  001412          BIT     #BIT3,RMCS10    ;WAS COMMAND A WRITE??
 321 060530  001406                          BEQ     210$            ;YES!!
 322 060532  112776  000316  000000  205$:   MOVB    #316,@(SP)      ;CHANGE ERROR NUMBER
 323 060540  042737  004000  001140          BIC     #WLE,$GDDAT
 324 060546  162716  000002          210$:   SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 325 060552  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 326 060554  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 327
 328 060560                          220$:
 329
 330                                 ;OMIT DATA ERROR CHECKS IF ANY PREVIOUS ERRORS HAVE BEEN DETECTED
 331 060560  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
 332 060564  105776  000000                  TSTB    @(SP)           ;WAS ERROR DETECTED??
 333 060570  001404                          BEQ     225$            ;NO - DO DATA CHECKS
 334 060572  162716  000004                  SUB     #4,(SP)         ;RESTORE SP
 335 060576  000137  061600                  JMP     340$            ;SKIP DATA CHECKS
 336 060602  162716  000004          225$:   SUB     #4,(SP)         ;RESTORE SP
 337
 338                                 ;CHECK HEADER ERRORS IF FUNCTION WAS NOT WRITE HEADER AND DATA, AND
 339                                 ;IF HEADER COMPARE IS NOT INHIBITED
 340 060606  013737  001412  062170          MOV     RMCS10,510$     ;STRIP AND STORE FUNCTION CODE
 341 060614  042737  177700  062170          BIC     #^CFNCMSK,510$
 342 060622  022737  000063  062170          CMP     #WHIGO,510$     ;WAS FUNCTION WRITE HEADER + DATA??
```

```
                                        BEQ     250$            ;YES - SKIP HEADER CHECKS
060632  032737  002000  001370          BIT     #HCI,RMOFI      ;WAS HCI SET??
060640  001106                          BNE     250$            ;YES - SKIP HEADER CHECKS

                                ;SEE IF ANY HEADER ERRORS ARE SET, I.E., "FER" OR "HCRC" OR "HCE"
060642  032737  000620  001352          BIT     #HCRC!FER.HCE,RMER1I
060650  001533                          BEQ     270$            ;NO ERRORS SET

                                ;REPORT HEADER CRC ERROR IF SET
060652  032737  000400  001352          BIT     #HCRC,RMER1I    ;WAS HCRC SET??
060660  001422                          BEQ     230$            ;NO!!
060662  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
060670  042737  000400  001140          BIC     #HCRC,$GDDAT
060676  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
060704  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
060710  112776  000317  000000          MOVB    #317,@(SP)      ;WRITE ERROR NUMBER
060716  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060722  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060724  000501                          BR      260$
060726                          230$:

                                ;REPORT FORMAT ERROR IF SET
060726  032737  000020  001352          BIT     #FER,RMER1I     ;WAS "FER" SET??
060734  001422                          BEQ     240$            ;NO!!
060736  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
060744  042737  000020  001140          BIC     #FER,$GDDAT
060752  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
060760  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
060764  112776  000320  000000          MOVB    #320,@(SP)      ;WRITE ERROR NUMBER
060772  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060776  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061000  000453                          BR      260$
061002                          240$:

                                ;REPORT HEADER COMPARE ERROR IF SET
061002  032737  000200  001352          BIT     #HCE,RMER1I     ;WAS "HCE" SET??
061010  001453                          BEQ     270$            ;NO!!
061012  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
061020  042737  000200  001140          BIC     #HCE,$GDDAT
061026  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
061034  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
061040  112776  000321  000000          MOVB    #321,@(SP)      ;WRITE ERROR NUMBER
061046  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061052  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061054  000425                          BR      260$
                                ;THERE SHOULD BE NO HEADER ERRORS BECAUSE
                                ;       .COMMAND WAS WRITE HEADER AND DATA, OR
                                ;       .HEADER COMPARE INHIBIT WAS SET
061056  032737  000620  001352  250$:   BIT     #HCE!FER!HCRC,RMER1I
061064  001425                          BEQ     270$            ;NO ERRORS WERE SET
061066  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
061074  042737  000620  001140          BIC     #HCE!FER!HCRC,$GDDAT
061102  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
061110  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
061114  112776  000322  000000          MOVB    #322,@(SP)      ;WRITE ERROR NUMBER
061122  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061126  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
```

```
                                        260$:   SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                                JMP     340$            ;OMIT FURTHER DATA CHECKS

                                        270$:

                                        ;IF COMMAND WAS A WRITE COMMAND, GO DO WRITE ERROR CHECKS, OTHERWISE
                                        ;DO READ ERROR CHECKS
                                                BIT     #BIT3,510$      ;WAS THIS A WRITE COMMAND?
                                                BNE     275$            ;NO..
                                                JMP     310$            ;GO DO WRITE STATUS CHECK
                                        275$:

                                        ;REPORT DATA CHECK IF SET
                                                BIT     #DCK,RMER1I     ;DATA CHECK ERROR??
                                                BEQ     290$            ;NO.!
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #DCK,$GDDAT
                                                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                                MOVB    #323,@(SP)      ;WRITE ERROR NUMBER
                                                BIT     #ECI,RMOFI      ;WAS ECC CORRECTION DISABLED??
                                                BNE     280$            ;YES!!
                                                MOVB    #324,@(SP)      ;CHANGE TO RECOVERABLE ERROR
                                                BIT     #ECH,RMER1I     ;IS ERROR RECOVERABLE??
                                                BNE     276$            ;NO !!
                                        ;DO NOT REPORT RECOVERABLE ERROR IF READ COMMAND
                                                BIT     #BIT4,510$      ;WAS THIS A READ COMMAND ??
                                                BEQ     280$            ;NO !!
                                                SUB     #4,(SP)         ;RESTORE SP
                                                BR      290$            ;SKIP ERROR - DATA WILL BE CORRECTED
                                        276$:   MOVB    #325,@(SP)      ;CHANGE TO NON RECOVERABLE
                                        280$:   SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR

                                        290$:

                                        ;REPORT DATA BUS PARITY ERROR IF SET, I.E., MDPE = 1
                                                BIT     #MDPE,RMCS2I    ;PARITY ERROR SET??
                                                BEQ     300$            ;NO'!
                                                MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #MDPE,$GDDAT
                                                MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                                MOVB    #326,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        300$:   JMP     340$            ;SKIP WRITE STATUS CHECK

                                        310$:

                                        ;TEST TO SEE THAT OFFSET MODE WAS RESET; REPORT ERROR IF 'OM'   1
                                                BIT     #OM,RMDSI       ;IS OFFSET ON??
                                                BEQ     320$            ;NO
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIC     #OM,$GDDAT
```

```
 457 061412  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
 458 061420  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 459 061424  112776  000327  000000          MOVB    #327,@(SP)      ;WRITE ERROR NUMBER IN CALL
 460 061432  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 461 061436  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 462 061440  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 463 061444                          320$:
 464
 465                                 ;TEST FOR DATA BUS PARITY ERROR; REPORT ERROR IF 'DPE'' = 1
 466 061444  032737  000010  001400          BIT     #DPE,RMER2I     ;DATA PARITY ERROR??
 467 061452  001423                          BEQ     330$            ;NO!!
 468 061454  013737  001400  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
 469 061462  042737  000010  001140          BIC     #DPE,$GDDAT
 470 061470  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
 471 061476  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 472 061502  112776  000330  000000          MOVB    #330,@(SP)      ;WRITE ERROR NUMBER
 473 061510  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 474 061514  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 475 061516  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 476 061522                          330$:
 477
 478                                 ;TEST FOR WRITE CLOCK FAILURE; REPORT ERROR IF 'WCF'' - 1
 479 061522  032737  000040  001352          BIT     #WCF,RMER1I     ;IS 'WCF'' SET??
 480 061530  001423                          BEQ     340$            ;NO!!
 481 061532  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 482 061540  042737  000040  001140          BIC     #WCF,$GDDAT
 483 061546  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 484 061554  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 485 061560  112776  000331  000000          MOVB    #331,@(SP)      ;WRITE ERROR NUMBER
 486 061566  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 487 061572  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 488 061574  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 489 061600                          340$:
 490
 491                                 ;REPORT 'DATA LATE'' ERROR IF 'DLT'' = 1
 492 061600  032737  100000  001346          BIT     #DLT,RMCS2I     ;IS 'DLT'' SET??
 493 061606  001423                          BEQ     350$            ;NO!!
 494 061610  013737  001346  001140          MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
 495 061616  042737  100000  001140          BIC     #DLT,$GDDAT
 496 061624  013737  001346  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
 497 061632  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 498 061636  112776  000332  000000          MOVB    #332,@(SP)      ;WRITE ERROR NUMBER
 499 061644  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 500 061650  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 501 061652  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 502 061656                          350$:
 503                                 ;LOOK FOR UNEXPECTED CHANGES IN DRIVE STATUS
 504 061656  013746  001350                  MOV     RMDSI,-(SP)     ;STACK DRIVE STATUS
 505 061662  042716  147677                  BIC     #^C<PIP!MOL!VV>,(SP) ;CLEAR DONT CARES
 506 061666  022726  010100                  CMP     #MOL!VV,(SP)+   ;IS DRIVE STATUS OK??
 507 061672  001522                          BEQ     380$            ;YES!.
 508
 509                                 ;REPORT ERROR IF POSITIONING IN PROGRESS AND NO SEEK INCOMPLETE ERROR,
 510                                 ;I.E. PIP - 1 AND SKI   0
 511 061674  032737  020000  001350          BIT     #PIP,RMDSI      ;IS 'PIP'' SET??
 512 061702  001430                          BEQ     360$            ;NO.!
 513 061704  032737  040000  001400          BIT     #SKI,RMER2I     ;WAS ''SKI'' ERROR REPORTED??
```

```
514 061712  001024                          BNE     360$            ;YES-DONT REPORT PIP
515 061714  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
516 061722  042737  020000  001140          BIC     #PIP,$GDDAT
517 061730  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
518 061736  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
519 061742  112776  000333  000000          MOVB    #333,@(SP)      ;WRITE ERROR NUMBER
520 061750  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
521 061754  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
522 061756  162716  000010                  SUB     #10.(SP)        ;MOVE SP TO NO ERROR
523 061762  000240                          NOP
524 061764                          360$:
525
526                                 ;REPORT ERROR IF MEDIUM IS NOT ON LINE AND OPI ERROR WAS NOT
527                                 ;REPORTED, I.E., MOL = OPI = 0
528 061764  032737  010000  001350          BIT     #MOL,RMDSI      ;IS MEDIUM ON LINE??
529 061772  001027                          BNE     370$            ;YES!!
530 061774  032737  020000  001352          BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
531 062002  001023                          BNE     370$            ;YES..
532 062004  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
533 062012  052737  010000  001140          BIS     #MOL,$GDDAT
534 062020  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
535 062026  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
536 062032  112776  000334  000000          MOVB    #334,@(SP)      ;WRITE ERROR NUMBER
537 062040  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
538 062044  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
539 062046  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
540 062052                          370$:
541
542                                 ;REPORT ERROR IF VOLUME IS NOT VALID AND "IVC" ERROR WAS NOT
543                                 ;REPORTED, I.E., VV = IVC = 0
544 062052  032737  000100  001350          BIT     #VV,RMDSI       ;IS VOLUME VALID??
545 062060  001027                          BNE     380$            ;YES!!
546 062062  032737  010000  001400          BIT     #IVC,RMER2I     ;WAS IVC ERROR REPORTED??
547 062070  001033                          BNE     390$            ;YES!!
548 062072  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
549 062100  052737  000100  001140          BIS     #VV,$GDDAT
550 062106  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
551 062114  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
552 062120  112776  000335  000000          MOVB    #335,@(SP)      ;WRITE ERROR NUMBER
553 062126  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
554 062132  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
555 062134  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
556 062140                          380$:
557
558                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS FOUND
559 062140  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
560 062144  105776  000000                  TSTB    @(SP)           ;ANY ERROR??
561 062150  001403                          BEQ     390$            ;NO!!
562 062152  062716  000004                  ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
563 062156  000402                          BR      400$
564 062160  162716  000004          390$:   SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
565
566 062164  000207                  400$:   RTS     PC              ;RETURN TO USER
567
568 062166  000000                  500$:   .WORD                   ;ERROR FLAGS
569 062170  000000                  510$:   .WORD                   ;TEMPORARY STORAGE
```

```
                                .SBTTL  STATIC DRIVE STATUS CHECK SUBROUTINE
                                
                                ;THIS SUBROUTINE LOOKS FOR UNEXPECTED CHANGES IN DRIVE
                                ;STATUS, SUCH AS THE DRIVE LOSING VOLUME VALID.  THE SUBROUTINE
                                ;CAN BE USED BY HOUSEKEEPING AND OTHER COMMANDS DURING WHICH THERE
                                ;SHOULD NOT BE ANY DRIVE ERRORS OR CHANGES IN STATE.
                                
                                ;THE FOLLOWING CONDITIONS ARE TESTED AND REPORTED AS ERRORS
                                ;IF TRUE:
                                
                                ;       .MOL = 0, INDICATES DRIVE WENT OFFLINE, NOTE
                                ;THAT MOL IS ASSUMED TO HAVE BEEN SET
                                ;       .VV = 0, INDICATES THE DRIVE LOST VOLUME VALID
                                ;       .PIP = 1, INDICATES THAT THE DRIVE IS OFF CYLINDER
                                ;       .SKI = 1, INDICATES THE DRIVE HAS AN UNEXPECTED SKI ERROR
                                ;       .DVC = 1, INDICATES AN UNEXPECTED DEVICE FAULT
                                
                                ;THE SUBROUTINE IS CALLED AFTER STORING STATUS IN THE GET BUFFER.
                                
                                ;(1)    JSR     PC,STCDRVSTS
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS
                                
062172                          STCDRVSTS:
                                
                                ;CLEAR USER'S ERROR CALL
062172  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
062176  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
062202  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR RETURN
                                ;SEE IF 'MOL' = 'VV' = 1, AND 'PIP' = 0
062206  013746  001350                  MOV     RMDSI,-(SP)     ;PUT DRIVE STATUS ON STACK
062212  042716  147677                  BIC     #^C<PIP!MOL!VV>,(SP)
062216  022726  010100                  CMP     #MOL!VV,(SP)+   ;ARE MOL,VV AND PIP O.K.??
062222  001524                          BEQ     30$             ;YES!!
                                
                                ;REPORT AN ERROR IF MOL = 0 AND 'OPI' = 0
062224  032737  010000  001350          BIT     #MOL,RMDSI      ;IS MOL ON ??
062232  001030                          BNE     10$             ;YES!!
062234  032737  020000  001352          BIT     #OPI,RMER1I     ;WAS 'OPI' SET??
062242  001024                          BNE     10$             ;YES-DONT REPORT 'MOL'  0
062244  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
062252  052737  010000  001140          BIS     #MOL,$GDDAT
062260  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
062266  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
062272  112776  000207  000000          MOVB    #207,@(SP)      ;WRITE ERROR NUMBER IN CALL
062300  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
062304  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
062306  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
062312  000240                          NOP
062314                          10$:
                                
                                ;REPORT AN ERROR IF VOLUME VALID IS NOW ZERO AND 'IVC'  0
062314  032737  000100  001350          BIT     #VV,RMDSI       ;IS 'VV' = 0??
062322  001030                          BNE     20$             ;NO.
```

```
 58 062324  032737  010000  001400          BIT     #IVC,RMER2I     ;WAS ''IVC'' SET??
 59 062332  001024                          BNE     20$             ;YES-DONT REPORT 'VV'   0
 60 062334  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
 61 062342  052737  000100  001350          BIS     #VV,RMDSI
 62 062350  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
 63 062356  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 64 062362  112776  000210  000000          MOVB    #210,@(SP)      ;WRITE ERROR NUMBER IN CALL
 65 062370  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 66 062374  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 67 062376  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 68 062402  000240                          NOP
 69 062404                          20$:
 70
 71                                 ;REPORT AN ERROR IF DRIVE IS OFF CYLINDER AND ''SKI'' - 0
 72 062404  032737  020000  001350          BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
 73 062412  001430                          BEQ     30$             ;NO.!
 74 062414  032737  040000  001400          BIT     #SKI,RMER2I     ;WAS ''SKI'' SET??
 75 062422  001024                          BNE     30$             ;YES-DONT REPORT 'PIP'' - 1
 76 062424  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
 77 062432  042737  020000  001140          BIC     #PIP,$GDDAT
 78 062440  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
 79 062446  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 80 062452  112776  000211  000000          MOVB    #211,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
 81 062460  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 82 062464  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 83 062466  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 84 062472  000240                          NOP
 85 062474                          30$:
 86
 87                                 ;SEE IF ''SKI'' = 'DVC'' = 0
 88 062474  013746  001400                  MOV     RMER2I,-(SP)    ;PUT ERROR REG 2 ON STACK
 89 062500  042726  137577                  BIC     #^C<SKI!DVC>,(SP)+
 90 062504  001460                          BEQ     60$             ;BRANCH IF NO ERROR
 91 062506                          40$:
 92
 93                                 ;REPORT AN ERROR IF THERE IS A DEVICE FAULT
 94 062506  032737  000200  001400          BIT     #DVC,RMER2I     ;ANY DEVICE FAULT??
 95 062514  001424                          BEQ     50$             ;NO!!
 96 062516  013737  001400  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
 97 062524  042737  000200  001140          BIC     #DVC,$GDDAT
 98 062532  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
 99 062540  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S CALL
100 062544  112776  000212  000000          MOVB    #212,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
101 062552  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
102 062556  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
103 062560  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
104 062564  000240                          NOP
105 062566                          50$:
106
107                                ;REPORT AN ERROR IF ''SKI'' = 1
108 062566  032737  040000  001400          BIT     #SKI,RMER2I     ;IS THERE A SEEK INCOMPLETE ERROR
109 062574  001424                          BEQ     60$             ;NO!!
110 062576  013737  001400  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
111 062604  042737  040000  001140          BIC     #SKI,$GDDAT
112 062612  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
113 062620  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
114 062624  112776  000213  000000          MOVB    #213,@(SP)      ;WRITE ERROR NUMBER IN USER'S ERROR CALL
```

```
115 062632  162716  000002          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
116 062636  004736                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
117 062640  162716  000010          SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
118 062644  000240                  NOP
119 062646                  60$:
120
121                         ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
122 062646  062716  000004          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
123 062652  105776  000000          TSTB    @(SP)           ;WAS AN ERROR DETECTED??
124 062656  001403                  BEQ     70$             ;NO!.
125 062660  062716  000004          ADD     #4,(SP)         ;YES - MOVE SP TO USER'S ERROR RETURN
126 062664  000402                  BR      80$
127 062666  162716  000004  70$:    SUB     #4,(SP)         ;NO - MOVE SP TO NO ERROR RETURN
128 062672  000240  80$:    NOP
129 062674  000207                  RTS     PC              ;RETURN TO USER
```

```
                        .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

062676                  $SAVREG:
062676  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
062700  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
062702  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
062704  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
062706  010446                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
062710  010546                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
062712  016646  000022         MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
062716  016646  000022         MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
062722  016646  000022         MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
062726  016646  000022         MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
062732  000002                 RTI

                        ;*RESTORE R0-R5
                        ;*CALL:
                        ;*      RESREG
062734                  $RESREG:
062734  012666  000022         MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
062740  012666  000022         MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
062744  012666  000022         MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
062750  012666  000022         MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
062754  012605                 MOV     (SP)+,R5        ;;POP STACK INTO R5
062756  012604                 MOV     (SP)+,R4        ;;POP STACK INTO R4
062760  012603                 MOV     (SP)+,R3        ;;POP STACK INTO R3
062762  012602                 MOV     (SP)+,R2        ;;POP STACK INTO R2
062764  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
062766  012600                 MOV     (SP)+,R0        ;;POP STACK INTO R0
062770  000002                 RTI
```

```
                                .SBTTL  BINARY TO ASCII AND TYPE ROUTINE

                                ;;*********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
                                ;*BINARY-ASCII NUMBER AND TYPE IT.
                                ;*CALL:
                                ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
                                ;*      TYPBN                   ;;TYPE IT

062772  010146                 $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
062774  016601  000006                 MOV     6(SP),R1        ;;GET THE INPUT NUMBER
063000  000261                         SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
063002  112737  000060  063044 1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
063010  006101                         ROL     R1              ;;GET THIS BIT
063012  001406                         BEQ     2$              ;;DONE?
063014  105537  063044                 ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
063020  104401  063044                 TYPE    ,$BIN           ;;GO TYPE THIS BIT
063024  000241                         CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
063026  000765                         BR      1$              ;;GO DO THE NEXT BIT
063030  012601                 2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
063032  016666  000002  000004         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
063040  012616                         MOV     (SP)+,(SP)
063042  000002                         RTI                     ;;RETURN TO USER
063044      000     000         $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
```

```
                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                        ;;*****************************************************************
                        ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                        ;;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                        ;;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                        ;;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                        ;;*REPLACED WITH SPACES.
                        ;;*CALL:
                        ;;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                        ;;*      TYPDS                   ;;GO TO THE ROUTINE

063046                  $TYPDS:
063046  010046          1$:     MOV     R0,-(SP)        ;;PUSH R0 ON STACK
063050  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
063052  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
063054  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
063056  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
063060  012746  020200          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
063064  016605  000020          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
063070  100004                  BPL     1$              ;;BR IF INPUT IS POS.
063072  005405                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
063074  112766  000055  000001  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
063102  005000          2$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
063104  012703  063262          MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
063110  112723  000040          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
063114  005002          3$:     CLR     R2              ;;CLEAR THE BCD NUMBER
063116  016001  063252          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
063122  160105          4$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
063124  002402                  BLT     4$              ;;BR IF DONE
063126  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
063130  000774                  BR      3$
063132  060105          5$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
063134  005702                  TST     R2              ;;CHECK IF BCD DIGIT 0
063136  001002                  BNE     5$              ;;FALL THROUGH IF 0
063140  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
063142  100407                  BMI     7$              ;;BR IF YES
063144  106316          6$:     ASLB    (SP)            ;;MSD?
063146  103003                  BCC     6$              ;;BR IF NO
063150  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
063156  052702  000060          BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
063162  052702  000040          BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
063166  110223          7$:     MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
063170  005720                  TST     (R0)+           ;;JUST INCREMENTING
063172  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
063176  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
063200  003002                  BGT     8$              ;;GO TO EXIT
063202  010502                  MOV     R5,R2           ;;GET THE LSD
063204  000764                  BR      6$              ;;GO CHANGE TO ASCII
063206  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
063210  100003                  BPL     9$              ;;BR IF NO
063212  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
063220  105013          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
063222  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
063224  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
063226  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
063230  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
063232  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
063234  104401  063262                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
063240  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
063246  012616                          MOV     (SP)+,(SP)
063250  000002                          RTI                     ;;RETURN TO USER
063252  023420                  $DTBL:  10000.
063254  001750                          1000.
063256  000144                          100.
063260  000012                          10.
063262                          $DBLK:  .BLKW   4
```

```
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

063272  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
063276  116637  000001  063515          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
063304  112637  063517                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
063310  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
063314  000406                          BR      $TYPON
063316  112737  000001  063515  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
063324  112737  000006  063517          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
063332  112737  000005  063514  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
063340  010346                          MOV     R3,-(SP)        ;;SAVE R3
063342  010446                          MOV     R4,-(SP)        ;;SAVE R4
063344  010546                          MOV     R5,-(SP)        ;;SAVE R5
063346  113704  063517                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
063352  005404                          NEG     R4
063354  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX  ALLOWED
063360  110437  063516                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
063364  113704  063515                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
063370  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
063374  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
063376  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
063400  000404                          BR      3$              ;;GO DO MSB
063402  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
063404  006105                          ROL     R5
063406  006105                          ROL     R5
063410  010503                          MOV     R5,R3
063412  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
063414  105337  063516                  DECB    $OMODE          ;;TYPE THIS DIGIT?
063420  100016                          BPL     7$              ;;BR IF NO
063422  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
063426  001002                          BNE     4$              ;;TEST FOR 0
063430  005704                          TST     R4              ;;SUPPRESS THIS 0?
063432  001403                          BEQ     5$              ;;BR IF YES
063434  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
```

```
063436  052703  000060            BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
063442  052703  000040      5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
063446  110337  063512            MOVB    R3,8$           ;;SAVE FOR TYPING
063452  104401  063512            TYPE    ,8$             ;;GO TYPE THIS DIGIT
063456  105337  063514      7$:   DECB    $OCNT           ;;COUNT BY 1
063462  003347                    BGT     2$              ;;BR IF MORE TO DO
063464  002402                    BLT     6$              ;;BR IF DONE
063466  005204                    INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
063470  000744                    BR      2$              ;;GO DO THE LAST DIGIT
063472  012605      6$:   MOV     (SP)+,R5        ;;RESTORE R5
063474  012604            MOV     (SP)+,R4        ;;RESTORE R4
063476  012603            MOV     (SP)+,R3        ;;RESTORE R3
063500  016666  000002  000004      MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
063506  012616            MOV     (SP)+,(SP)
063510  000002            RTI                     ;;RETURN
063512     000      8$:   .BYTE   0               ;;STORAGE FOR ASCII DIGIT
063513     000            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
063514     000      $OCNT:.BYTE   0               ;;OCTAL DIGIT COUNTER
063515     000      $OFILL:.BYTE  0               ;;ZERO FILL SWITCH
063516  000000      $OMODE:.WORD  0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .SBTTL  TYPE ROUTINE

                                ;;*********************************************************************
                                ;;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;;*
                                ;;*CALL:
                                ;;*1) USING A TRAP INSTRUCTION
                                ;;*     TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;;*OR
                                ;;*     TYPE
                                ;;*     MESADR
                                ;;*

063520  105737  001173          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
063524  100002                          BPL     1$              ;;BR IF YES
063526  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
063530  000430                          BR      3$              ;;LEAVE
063532  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
063534  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
063540  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
063546  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
063550  132737  000100  001243          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
063556  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
063560  010037  063570                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
063564  004737  070166                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
063570  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
063572  132737  000040  001243  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
063600  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
063602  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
063604  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
063606  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
063610  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
063612  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
063616  000002                          RTI                     ;;RETURN
063620  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
063624  001430                          BEQ     8$
063626  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
063632  001006                          BNE     5$
063634  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
063636  104401                          TYPE                    ;;TYPE A CR AND LF
063640  001217                          $CRLF
063642  105037  064050                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
063646  000755                          BR      2$              ;;GET NEXT CHARACTER
063650  004737  063732          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
063654  123726  001172          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
063660  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
063662  013746  001170                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
063666  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
063672  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
063674  004737  063732                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
063700  105337  064050                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
063704  000770                          BR      7$              ;;LOOP
```

```
                                        ;HORIZONTAL TAB PROCESSOR

063706  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
063712  004737  063732          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
063716  132737  000007  064050          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
063724  001372                          BNE     9$              ;;TAB STOP
063726  005726                  .       TST     (SP)+           ;;POP SPACE OFF STACK
063730  000724                          BR      2$              ;;GET NEXT CHARACTER
063732                          $TYPEC:
063732  105777  115222                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
063736  100022                          BPL     10$             ;;BR IF NOT
063740  017746  115216                  MOV     @$TKB,-(SP)     ;;GET CHAR
063744  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
063750  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
063754  001012                          BNE     102$            ;;BR IF NOT
063756                          101$:
063756  105777  115176                  TSTB    @$TKS           ;;WAIT FOR CHAR
063762  100375                          BPL     101$
063764  117716  115172                  MOVB    @$TKB,(SP)      ;;GET CHAR
063770  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
063774  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
064000  001366                          BNE     101$            ;;BR IF NOT
064002                          102$:
064002  005726                          TST     (SP)+           ;;FIX STACK
064004                          10$:
064004  105777  115154                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
064010  100375                          BPL     10$
064012  116677  000002  115146          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
064020  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
064026  001003                          BNE     1$              ;;BRANCH IF NO
064030  105037  064050                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
064034  000406                          BR      $TYPEX          ;;EXIT
064036  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
064044  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
064046  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
064050  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
064052  000207          $TYPEX: RTS     PC
```

```
                                .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

064054                          $SCOPE:
064054  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
064056  004737  064612                   JSR     PC,STOP
064062  032777  040000  115064  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
064070  001402                          BEQ     9$              ;;NO IF SW14=0
064072  000137  064522                   JMP     $OVER           ;;JUMP OVER SCOPE ROUTINE
064076                          9$:
                                ;#####START OF CODE FOR THE XOR TESTER#####
064076  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
064100  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
064104  012737  064124  000004           MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
064112  005737  177060                   TST     @#177060        ;;TIME OUT ON XOR?
064116  012637  000004                   MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
064122  000561                          BR      $SVLAD          ;;GO TO THE NEXT TEST
064124  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
064126  012637  000004                   MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
064132  000521                          BR      7$              ;;LOOP ON THE PRESENT TEST
064134                          6$:;#####END OF CODE FOR THE XOR TESTER#####
064134  032777  000400  115012           BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
064142  001421                          BEQ     2$              ;;BR IF NO
064144  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
064146  117716  115002                   MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
064152  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
064154  022716  000025                   CMP     #25,(SP)        ;;CHECK THE NUMBER IN THE SWR
064160  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
064162  011637  001116                   MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
064166  005316                          DEC     (SP)            ;;BACKUP BY ONE
064170  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
064172  062716  064540                   ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
064176  013637  001122                   MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
064202  000547                          BR      $OVER           ;;GO LOOP ON THE TEST
064204  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
064206  105737  001117          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
064212  001502                          BEQ     3$              ;;BR IF NO
064214  022737  177777  065230           CMP     #-1,CPSAVE      ;;SEE IF TIMFOUT WAS PREVIOUSLY RECORDED
064222  001455                          BEQ     2003$           ;;KICK AROUND ROUTINE IF SO
064224  013746  000004                   MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
064230  012737  064246  000004           MOV     #2000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
064236  013737  177766  065230           MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
064244  000406                          BR      2001$
064246  012737  177777  065230  2000$:  MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
064254  012716  064262                   MOV     #2001$,(SP)     ;;SETUP RETURN ADDRESS
```

```
064260  000002                          RTI
064262  012637  000004         2001$:   MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

064266  022737  177777  065230 2002$:   CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
064274  001430                          BEQ     2003$           ;;BRANCH IF SO
064276  032737  000001  065230         BIT     #BIT00,CPSAVE   ;;SEE IF THE POWER MONITOR BIT IS ON
064304  001424                          BEQ     2003$           ;;BRANCH TO CONTINUE ROUTINE IF CLEAR
064306  042737  000001  177766         BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND TO BE SET
064314  013746  001154                  MOV     SWR,-(SP)       ;;SAVE SWR ADDRESS
064320  017646  000000                  MOV     @(SP),-(SP)     ;;SAVE SWR VALUE
064324  012737  000176  001154         MOV     #176,SWR        ;;GET SOFTWARE SWR ADDRESS
064332  011677  114616                  MOV     (SP),@SWR       ;;GET CURRENT SWR VALUE
064336  042777  001000  114610         BIC     #BIT09,@SWR     ;;DON'T ALLOW LOOP ON ERROR ON THIS ERROR
064344  104177                          EMT     177             ;;CALL SPECIAL POWER FAIL BIT ERROR CALL
064346  012676  000000                  MOV     (SP)+,@(SP)     ;;RESTORE SWR TO ORIGINAL VALUE
064352  012637  001154                  MOV     (SP)+,SWR       ;;RESTORE SWR ADDRESS
064356                          2003$:
064356  123737  001131  001117         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
064364  101015                          BHI     3$              ;;BR IF NO
064366  032777  001000  114560         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
064374  001404                          BEQ     4$              ;;BR IF NO
064376  013737  001124  001122 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
064404  000446                          BR      $OVER
064406  105037  001117         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
064412  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
064416  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
064420  032777  004000  114526 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
064426  001011                          BNE     1$              ;;BR IF YES
064430  005737  001230                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
064434  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
064436  005237  001120                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
064442  023737  001206  001120         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
064450  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
064452  012737  000001  001120 1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
064460  013737  064536  001206         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
064466  105237  001116         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
064472  113737  001116  001226         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
064500  011637  001122                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
064504  011637  001124                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
064510  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
064514  112737  000001  001131         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
064522  013777  001116  114426 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
064530  013716  001122                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
064534  000002                          RTI                     ;;FIXES PS
064536  000005                  $MXCNT: 5.                      ;;MAX. NUMBER OF ITERATIONS
064540                          $SW08TBL:
        000025                  .REPT   $TN-1
064540  010070                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
064542  010270                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
064544  011276                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
064546  012254                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
064550  013450                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
064552  014456                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
064554  015434                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
064556  016632                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
064560  017636                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
064562  021062                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
```

```
064564  022270                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
064566  023312                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
064570  025160                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
064572  026564                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
064574  027604                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
064576  030522                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
064600  031560                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
064602  032564                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
064604  033572                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
064606  034600                          .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
064610  035626                          .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25

                        ;DROP PRIORITY TO ALLOW CONSOLE INTERRUPT

064612                  STOP:
064612  012746  000140          MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
064616  012746  064624          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
064622  000002                  RTI                     ;;POP NEW PC AND PS
064624                  64$:

                        ;RAISE PRIORITY TO INHIBIT CONSOLE INTERRUPT

064624  012746  000240          MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
064630  012746  064636          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
064634  000002                  RTI                     ;;POP NEW PC AND PS
064636                  65$:
064636  000207                  RTS     PC              ;RETURN
```

```
                        .SBTTL  ERROR HANDLER ROUTINE

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

064640  105037  065232         $ERROR: CLRB    IBSAVE          ;;CLEAR THE ITEM BYTE SAVE LOCATION
064644  104410                         CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
064646  105237  001117         7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
064652  001775                         BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
064654  013777  001116  114274         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
064562  032777  002000  114264         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
064670  001402                         BEQ     1$              ;;NO - SKIP
064672  104401  001212                 TYPE    ,$BELL          ;;RING BELL
064676  005237  001126         1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
064702  011637  001132                 MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
064706  162737  000002  001132         SUB     #2,$ERRPC
064714  117737  114212  001130         MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
064722  032777  001000  114224         BIT     #BIT09,@SWR     ;;SEE IF LOOP ON ERROR IS SET
064730  001060                         BNE     1004$           ;;BRANCH AROUND ROUTINE IF SO
064732  122737  000177  001130         CMPB    #177,$ITEMB     ;;SEE IF THIS IS THE POWER FAIL CALL
064740  001454                         BEQ     1004$           ;;BRANCH AROUND ROUTINE IF IT IS
064742  105737  065232                 TSTB    IBSAVE          ;;SEE IF THIS IS THE 2ND ERROR CALL IN THIS ROUTINE
064746  001047                         BNE     1003$           ;;BRANCH IF SO
064750  022737  177777  065230         CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
064756  001445                         BEQ     1004$           ;;BRANCH IF SO
064760  013746  000004                 MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
064764  012737  065002  000004         MOV     #1000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
064772  013737  177766  065230         MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
065000  000406                         BR      1001$
065002  012737  177777  065230  1000$: MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
065010  012716  065016                 MOV     #1001$,(SP)     ;;SETUP RETURN ADDRESS
065014  000002                         RTI
065016  012637  000004         1001$: MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

065022  022737  177777  065230  1002$: CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
065030  001420                         BEQ     1004$           ;;BRANCH IF SO
065032  032737  000001  065230         BIT     #BIT00,CPSAVE   ;;SEE IF POWER MONITOR BIT IS SET IN CPU ERR REG
065040  001414                         BEQ     1004$           ;;BRANCH IF OK
065042  042737  000001  177766         BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND SET
065050  113737  001130  065232         MOVB    $ITEMB,IBSAVE   ;;MAKE IBSAVE NON-ZERO FOR DUAL ERROR CALL
065056  112737  000177  001130         MOVB    #177,$ITEMB     ;;SET $ITEMB TO SPECIAL POWER FAIL POINTER
065064  000402                         BR      1004$           ;;BRANCH OVER IBSAVE CLEARING

065066  105037  065232         1003$: CLRB    IBSAVE          ;;CLEAR IBSAVE SO 2ND TIME THROUGH EXITS
065072                         1004$:
065072  032777  020000  114054         BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
065100  001004                         BNE     20$             ;;SKIP TYPEOUTS
065102  004737  065234                 JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
```

```
065106  104401  001217                  TYPE    ,$CRLF
065112                          20$:
065112  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
065120  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
065122  113737  001130  065134          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
065130  004737  070176                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
065134     000                  21$:    .BYTE   0
065135     000                          .BYTE   0
065136  000777                  22$:    BR      22$             ;;APT ERROR LOOP
065140  105737  065232          2$:     TSTB    IBSAVE          ;;SEE IF IBSAVE IS LOADED
065144  001005                          BNE     3$              ;;BRANCH IF NOT - NO HALT ON PWR MON BIT ERROR
065146  005777  114002                  TST     @SWR            ;;HALT ON ERROR
065152  100002                          BPL     3$              ;;SKIP IF CONTINUE
065154  000000                          HALT                    ;;HALT ON ERROR.
065156  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
065160                          3$:
065160  032777  001000  113766          BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
065166  001402                          BEQ     4$              ;;BR IF NO
065170  013716  001124                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
065174  005737  001210          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
065200  001402                          BEQ     5$              ;;BR IF NONE
065202  013716  001210                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
065206                          5$:
065206  022737  037242  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
065214  001001                          BNE     6$              ;;BRANCH IF NO
065216  000000                          HALT                    ;;YES
065220                          6$:
065220  105737  065232          7$:     TSTB    IBSAVE          ;;SEE IF ITEM BYTE SAVE LOCATION HAS AN ERROR CALL
065224  001210                          BNE     7$              ;;BRANCH BACK TO CALL ORIGINAL ERROR
065226  000002                          RTI                     ;;RETURN
065230  000000                  CPSAVE: .WORD   0               ;;LOCATION TO SAVE CPU ERROR REG CONTENTS
065232  000000                  IBSAVE: .WORD   0               ;;LOCATION TO SAVE ITEM BYTE
```

```
                                        .SBTTL  ERROR TYPEOUT ROUTINE
                                ;
                                ;*THE ERROR TYPEOUT ROUTINE ASSEMBLES AND PRINTS INFORMATION
                                ;*REGARDING THE DETECTION OF AN ERROR AS FOLLOWS:
                                ;*
                                ;*      .UNIT NUMBER, DRIVE TYPE, TEST NUMBER, ERROR NUMBER AND
                                ;*PROGRAM COUNTER ARE PRINTED ON THE FIRST LINE;
                                ;*      .ERROR MESSAGE IS ASSEMBLED, FORMATTED AND PRINTED ON
                                ;*ONE OR MORE SUCCEEDING LINES;
                                ;*      .PAIRED LINES OF ERROR HEADERS AND ERROR DATA ARE PRINTED
                                ;*AFTER THE ERROR MESSAGE.
                                ;
 065234  104414                 ERRTYP: SAVREG
 065236  032777  020000  113710         BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
 065244  001402                         BEQ     1$              ;NO..
 065246  000137  066074                 JMP     27$             ;YES!.
                                ;
                                ;TYPE UNIT NUMBER, DRIVE TYPE, TEST NUMBER, ERROR NUMBER, AND
                                ;PROGRAM COUNTER
 065252  104401  001217         1$:     TYPE    ,$CRLF
 065256  104401  066110                 TYPE    ,ERTY00         ;TYPE 'DRV#''
 065262  013746  001234                 MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                 ;;TYPE DRIVE NUMBER
 065266  104403                         TYPOS                   ;;GO TYPE--OCTAL ASCII
 065270     003                         .BYTE   3               ;;TYPE 3 DIGIT(S)
 065271     000                         .BYTE   0               ;;SUPPRESS LEADING ZEROS
                                ;
                                ;TYPE 'DRIVE TYPE' RM05, RM03 OR RM02 FOR UNIT UNDER TEST
 065272  013700  001276                 MOV     $BASE,R0        ;GET RM BASE ADDRESS
 065276  016000  000026                 MOV     RMDT(R0),R0     ;GET DRIVE TYPE REGISTER
 065302  042700  177740                 BIC     #177740,R0      ;SAVE DRIVE TYPE BITS AND
 065306  012737  071453  065360         MOV     #$RM03,3$       ;GET ASCII DRIVE TYPE
 065314  022700  000024                 CMP     #24,R0          ;IS DEVICE AN RM03 ?
 065320  001414                         BEQ     2$              ;YES !!
                                ;
 065322  012737  071446  065360         MOV     #$RM02,3$       ;SAVE ASCII DRIVE TYPE
 065330  022700  000025                 CMP     #25,R0          ;IS DEVICE AN RM02 ?
 065334  001406                         BEQ     2$              ;YES !!
                                ;
 065336  012737  071460  065360         MOV     #$RM05,3$       ;SAVE ASCII DRIVE TYPE
 065344  022700  000027                 CMP     #27,R0          ;IS DEVICE AN RM05 ?
 065350  001004                         BNE     4$              ;NO !!
 065352  104401  066145         2$:     TYPE    ,ERTY05         ;TYPE '' - ''
 065356  104401                         TYPE                    ;TYPE DRIVE TYPE
 065360  000000                 3$:     .WORD   0               ;DRIVE TYPE MESSAGE IS STORED HERE
                                ;
                                ;TYPE TEST NUMBER, ERROR NUMBER AND PROGRAM COUNTER
 065362  005037  066100         4$:     CLR     TSTNMB          ;LOAD TEST NUMBER FOR
 065366  013737  001226  066100         MOV     $TESTN,TSTNMB
 065374  104401  066115                 TYPE    ,ERTY01         ;TYPE ''TST#''
 065400  013746  066100                 MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                 ;;TYPE TEST NUMBER
 065404  104403                         TYPOS                   ;;GO TYPE--OCTAL ASCII
 065406     003                         .BYTE   3               ;;TYPE 3 DIGIT(S)
 065407     000                         .BYTE   0               ;;SUPPRESS LEADING ZEROS
 065410  005037  066102                 CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
 065414  113737  001130  066102         MOVB    $ITEMB,ERRNMB   ;TYPEOUT
```

ERROR TYPEOUT ROUTINE

```
 50 065422  001406                        BEQ     5$              ;SKIP IF NO ERROR CALLED
 51 065424  104401  066125                TYPE    ,ERTY02         ;TYPE "RR#"
 52 065430  013746  066102                MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                   ;;TYPE ERROR NUMBER
    065434  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
    065436     003                        .BYTE   3               ;;TYPE 3 DIGIT(S)
    065437     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 53 065440  104401  066134          5$:   TYPE    ,ERTY03         ;TYPE "PC="
 54 065444  013746  001132                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                   ;;TYPE PROGRAM COUNTER
    065450  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
    065452     006                        .BYTE   6               ;;TYPE 6 DIGIT(S)
    065453     001                        .BYTE   1               ;;TYPE LEADING ZEROS
 55
 56                                ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
 57 065454  005737  066102          6$:   TST     ERRNMB          ;WAS AN ERROR CALLED?
 58 065460  001002                        BNE     7$              ;BR IF YES
 59 065462  000137  066074                JMP     27$             ;NO--EXIT
 60
 61 065466  104401  001217          7$:   TYPE    ,$CRLF          ;YES-TYPE CRLF
 62 065472  105037  066106                CLRB    BOTFLG          ;CLEAR BOT FLAG
 63 065476  105037  066107                CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
 64 065502  013700  066102                MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
 65 065506  122700  000177                CMPB    #177,R0         ;SEE IF THIS ERROR CALL IS SPECIAL POWER FAIL CALL
 66 065512  001003                        BNE     8$              ;BRANCH IF NOT
 67 065514  012700  066152                MOV     #PFECH,R0       ;MOVE POWER FAIL ERROR CALL TABLE TO R0
 68 065520  000405                        BR      9$
 69 065522  006300                  8$:   ASL     R0              ;FOUR ENTRIES IN ERROR
 70 065524  006300                        ASL     R0              ;TABLE
 71 065526  006300                        ASL     R0
 72 065530  062700  001572                ADD     #$ERRTB-8.,R0
 73 065534  011001                  9$:   MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
 74                                                               ;TABLE
 75 065536  001507                        BEQ     19$             ;BRANCH IF NO ERROR MESSAGE
 76
 77                                ;TYPE THE ERROR MESSAGE
 78 065540  012102                 10$:   MOV     (R1)+,R2        ;R2-ADDRESS OF MESSAGE STRING
 79 065542  001505                        BEQ     19$             ;BRANCH IF END OF MESSAGE
 80 065544  010237  065712                MOV     R2,18$          ;LOAD ADDRESS OF STRING
 81 065550  005037  066104                CLR     BOTADR          ;CLEAR BOT ADDRESS
 82 065554  112203                 11$:   MOVB    (R2)+,R3        ;END OF STRING??
 83 065556  001454                        BEQ     17$             ;YES!!
 84 065560  122703  000015                CMPB    #CR,R3          ;CARRIAGE RETURN??
 85 065564  001003                        BNE     12$             ;NO!!
 86 065566  105037  066107                CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
 87 065572  000770                        BR      11$             ;GET NEXT CHARACTER
 88 065574  122703  000012          12$:  CMPB    #LF,R3          ;LINE FEED??
 89 065600  001765                        BEQ     11$             ;YES-GET NEXT CHARACTER
 90 065602  122703  000011                CMPB    #HT,R3          ;HORIZONTAL TAB??
 91 065606  001007                        BNE     14$             ;NO..
 92 065610  105237  066107          13$:  INCB    CHRCNT          ;ADJUST CHARACTER COUNT
 93 065614  132737  000007  066107        BITB    #7,CHRCNT
 94 065622  001372                        BNE     13$
 95 065624  000407                        BR      15$
 96 065626  105237  066107          14$:  INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
 97 065632  122703  000040                CMPB    #' ,R3          ;SPACE??
 98 065636  001002                        BNE     15$             ;NO.
```

```
 99 065640  010237  066104                 MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
100 065644  122737  000100  066107  15$:    CMPB    #64.,CHRCNT     ;END OF LINE??
101 065652  103340                         BHIS    11$             ;NO.!
102 065654  013704  066104                 MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
103 065660  001007                         BNE     16$             ;BRANCH IF SPACE DETECTED
104 065662  104401  001217                 TYPE    ,$CRLF          ;TYPE CRLF
105 065666  105037  066107                 CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
106 065672  013702  065712                 MOV     18$,R2          ;SET UP R2 FOR TESTING
107 065676  000726                         BR      11$
108 065700  105044                 16$:    CLRB    -(R4)           ;REPLACE SPACE
109 065702  112737  177777  066106         MOVB    #-1,BOTFLG      ;SET BOT FLAG
110 065710  104401                 17$:    TYPE                    ;TYPE ERROR MESSAGE STRING
111 065712  000000                 18$:    .WORD                   ;STRING ADDRESS GOES HERE
112 065714  105737  066106                 TSTB    BOTFLG          ;WAS STRING TRUNCATED??
113 065720  001707                         BEQ     10$             ;NO!!
114 065722  104401  001217                 TYPE    ,$CRLF          ;YES-TYPE CRLF
115 065726  105037  066106                 CLRB    BOTFLG          ;CLEAR BOT FLAG
116 065732  105037  066107                 CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
117 065736  013702  066104                 MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
118 065742  010237  065712                 MOV     R2,18$          ;SETUP 18$ FOR TYPING
119 065746  112742  000040                 MOVB    #' ,-(R2)       ;RESTORE SPACE
120 065752  105722                         TSTB    (R2)+           ;RESTORE R2
121 065754  000677                         BR      11$             ;TYPE REST OF STRING
122
123                                 ;TYPE ERROR HEADER AND ERROR DATA
124 065756  016001  000002         19$:    MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
125 065762  001444                         BEQ     27$             ;BRANCH IF NO HEADER
126 065764  104401  001217                 TYPE    ,$CRLF          ;(ASSUME NO DATA)
127 065770  016002  000004                 MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
128 065774  016003  000006                 MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
129 066000  012137  066010         20$:    MOV     (R1)+,21$       ;PUT HEADER ADDRESS FOR TYPE
130 066004  001433                         BEQ     27$             ;BRANCH IF END OF HEADERS
131                                                                 ;(ASSUME END OF DATA)
132 066006  104401                         TYPE
133 066010  000000                 21$.    .WORD   0               ;HEADER ADDRESS GOES HERE
134 066012  104401  001217                 TYPE    ,$CRLF
135 066016  005702                         TST     R2              ;DATA WITH HEADER??
136 066020  001767                         BEQ     20$             ;NO!!
137 066022  012204                         MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
138 066024  012305                         MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
139 066026  105725                 22$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
140 066030  100407                         BMI     24$             ;BINARY
141 066032  001403                         BEQ     23$             ;OCTAL
142 066034  013446                         MOV     @(R4)+,-(SP)    ;DECIMAL
143 066036  104405                         TYPDS
144 066040  000405                         BR      25$
145 066042  013446                 23$:    MOV     @(R4)+,-(SP)
146 066044  104402                         TYPOC
147 066046  000402                         BR      25$
148 066050  013446                 24$:    MOV     @(R4)+,-(SP)
149 066052  104406                         TYPBN
150 066054  005714                 25$:    TST     (R4)            ;MORE DATA??
151 066056  001403                         BEQ     26$             ;NO.!
152 066060  104401  066142                 TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
153 066064  000760                         BR      22$             ;AND CONTINUE
154 066066  104401  001217         26$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
155 066072  000742                         BR      20$             ;BEFORE NEXT HEADER
```

```
                        27$:    RESREG
                                RTS     PC

                        TSTNMB: .WORD   0               ;TEST NUMBER
                        ERRNMB: .WORD   0               ;ERROR NUMBER
                        BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
                        BOTFLG: .BYTE   0               ;BOT FLAG
                        CHRCNT: .BYTE   0               ;CHARACTER COUNT

                        ERTY00: .ASCIZ  @D'V#@
                        ERTY01: .ASCIZ  @, TEST#@
                        ERTY02: .ASCIZ  @, ERR#@
                        ERTY03: .ASCIZ  @, PC=@
                        ERTY04: .ASCIZ  @ @
                        ERTY05: .ASCIZ  @ - @
                                .EVEN
                        PFECH:  PFECH1,PFECH2,PFECH3,PFECH4 ;WORDS DEFINING TABLES BELOW
                        PFECH1: .+4,0
                                .ASCIZ  ?POWER MONITOR BIT IN CPU ERROR REGISTER FOUND SET?
                                .EVEN
                        PFECH2: .+4,0
                                .ASCIZ  ?CPUERREG?
                                .EVEN
                        PFECH3: .+2
                                .WORD   CPSAVE,0
                        PFECH4: .+2
                                .BYTE   0,0
```

```
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*******************************************************************
                                .ENABL  LSB
066300  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
066302  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
066304  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
066306                          $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        066307                  $TKQEND=.
                                .EVEN

                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIA.IZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN
                                ;
066310  005037  066300          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
066314  012737  066306  066302          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
066322  013737  066302  066304          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
066330  012737  066360  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
066336  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;'BR' LEVEL 4
066344  005777  112612                  TST     @$TKB           ;;CLEAR DONE FLAG
066350  012777  000100  112602          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
066356  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 (SHUT2)
                                ;
066360  117746  112576          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
066364  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
066370  021627  000021                  CMP     (SP),#$XON      ;;IS IT A RANDOM XON?
066374  001002                  BNE     30$             ;;BRANCH IF NO
066376  005726                  TST     (SP)+           ;;CLEAN RANDOM XON OFF STACK
066400  000002                  RTI                     ;;RETURN
066402                          30$:
066402  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
066406  001007                  BNE     1$              ;;BRANCH IF NO
066410  104401  067506                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
066414  004737  066310                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
066420  005726                  TST     (SP)+           ;;CLEAN UP STACK
066422  000137  067550                  JMP     SHUT2           ;;CONTROL C RESTART
066426  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
066432  001004                  BNE     2$              ;;BRANCH IF NO
066434  022737  000176  001154          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
066442  001500                  BEQ     6$              ;;GO TO SWR CHANGE

066444                          2$:
066444  022737  000001  066300          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
066452  001004                  BNE     3$              ;;BRANCH IF NO
066454  104401  001212                  TYPE    ,$BELL          ;;RING THE TTY BELL
```

```
066460  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
066462  000451                          BR      5$              ;;EXIT
066464  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
066470  001021                          BNE     32$             ;;BRANCH IF NO
066472  005077  112462                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
066476  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
066500  105777  112454          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
066504  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
066506  117746  112450                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
066512  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
066516  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
066522  001366                          BNE     31$             ;;BRANCH IF NO
066524  012777  000100  112426          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
066532  000002                          RTI                     ;;RETURN
066534  005237  066300          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
066540  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
066544  002405                          BLT     4$              ;;BRANCH IF YES
066546  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
066552  003002                          BGT     4$              ;;BRANCH IF YES
066554  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
066560  112677  177516          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
066564  005237  066302                  INC     $TKQIN          ;;UPDATE THE POINTER
066570  023727  066302  066307          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
066576  001003                          BNE     5$              ;;BRANCH IF NO
066600  012737  066306  066302          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
066606  000002                  5$:     RTI                     ;;RETURN

                                ;;*****************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                                ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
066610  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
066616  001124                          BNE     15$             ;;EXIT IF NOT
066620  105777  112334                  TSTB    @$TKS           ;;IS A CHAR WAITING?
066624  100121                          BPL     15$             ;;IF NOT, EXIT
066626  117746  112330                  MOVB    @$TKB,-(SP)     ;;YES
066632  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
066636  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
066642  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
                                                                ;;AND EXIT

                                ;;*****************************************************************
                                ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                                ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
066644  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
066652  001674                          BEQ     2$              ;;BRANCH IF YES
066654  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
066656  004737  066310                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
066662  005077  112272                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
066666  112737  000001  001151          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

066674  104401  067520                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
066700  104401  067525  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
066704  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
066710  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
066712  104401  067536                  TYPE    ,$MNFW          ;;PROMPT FOR NEW SWR
066716  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
066720  005046                          CLR     -(SP)           ;;THE NEW SWR
066722  105777  112232          7$:     TSTB    @$TKS           ;;CHAR THERE?
066726  100375                          BPL     7$              ;;IF NOT TRY AGAIN

066730  117746  112226                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
066734  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

066740  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
066744  001015                          BNE     9$              ;;BRANCH IF NOT
066746  104401  067506                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
066752  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
066756  123727  001151  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
066764  001003                          BNE     8$              ;;BRANCH IF NO
066766  012777  000100  112164          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
066774  000137  067550          8$:     JMP     SHUT2           ;;CONTROL-C RESTART


067000  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
067004  001005                          BNE     10$             ;;BRANCH IF NOT
067006  104401  067513                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
067012  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
067016  000737                          BR      19$             ;;LET'S TRY IT AGAIN


067020  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
067024  001022                          BNE     16$             ;;BRANCH IF NO
067026  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
067032  001403                          BEQ     11$             ;;BRANCH IF YES
067034  016677  000002  112112          MOV     2(SP),@SWR      ;;SAVE NEW SWR
067042  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
067046  104401  001217          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
067052  123727  001151  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
067060  001003                          BNE     15$             ;;BRANCH IF NOT
067062  012777  000100  112070          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
067070  000002                  15$:    RTI                     ;;RETURN
067072  004737  063732          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
067076  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
067102  002420                          BLT     18$             ;;BRANCH IF YES
067104  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
067110  003015                          BGT     18$             ;;BRANCH IF YES
067112  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
067116  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
067122  001403                          BEQ     17$             ;;BRANCH IF YES
067124  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
067126  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
067130  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
067132  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
067136  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
067142  000667                          BR      7$              ;;GET THE NEXT ONE
067144  104401  001216          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
067150  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
                                ;

067152  011646                 $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
067154  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
067162  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
067166  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
067170  012746  067176                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
067174  000002                          RTI                     ;;POP NEW PC AND PS
067176                          64$:
067176  005737  066300          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
067202  001775                          BEQ     1$
067204  005337  066300                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
067210  117766  177070  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
067216  005237  066304                  INC     $TKQOUT         ;;UPDATE THE POINTER
067222  023727  066304  066307          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
067230  001003                          BNE     2$              ;;BRANCH IF NO
067232  012737  066306  066304          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
067240  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

067242  010346                 $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
067244  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
067246  012703  067476          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
067252  022703  067506          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
067256  101456                          BLOS    4$              ;;BR IF YES
067260  104411                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
067262  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
067264  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
067270  001022                          BNE     5$              ;;BR IF NO
067272  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
067274  001007                          BNE     6$              ;;BR IF NO
067276  112737  000134  067474          MOVB    #'\,9$          ;;TYPE A BACK SLASH
067304  104401  067474                  TYPE    ,9$
067310  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
067314  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
067316  020327  067476                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
067322  103434                          BLO     4$              ;;BR IF YES
067324  111337  067474                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
067330  104401  067474                  TYPE    ,9$             ;;GO TYPE
067334  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
067336  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
067340  001406                          BEQ     7$              ;;BR IF NO
067342  112737  000134  067474          MOVB    #'\,9$          ;;TYPE A BACK SLASH
067350  104401  067474                  TYPE    ,9$
067354  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
067356  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
067362  001003                          BNE     8$              ;;BR IF NO
```

```
067364  104401  067513                  TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U''
067370  000726                          BR      1$              ;;GO START OVER
067372  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A ''^R''?
067376  001011                          BNE     3$              ;;BRANCH IF NO
067400  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
067402  104401  001217                  TYPE    ,$CRLF          ;;TYPE A ''CR'' & ''LF''
067406  104401  067476                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
067412  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
067414  104401  001216          4$:     TYPE    ,$QUES          ;;TYPE A '?'
067420  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
067422  111337  067474          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
067426  104401  067474                  TYPE    ,9$
067432  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
067436  001305                          BNE     2$              ;;LOOP IF NOT RETURN
067440  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
067444  104401  001220                  TYPE    ,$LF            ;;TYPE A LINE FEED
067450  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
067452  012603                          MOV     (SP)+,R3        ;;RESTORE R3
067454  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
067456  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
067464  012766  067476  000004          MOV     #$TTYIN,4(SP)
067472  000002                          RTI                     ;;RETURN
067474     000          9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
067475     000                  .BYTE   0               ;;TERMINATOR
067476                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
067506     136     103  015     $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL ''C''
067513     136     125  015     $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U''
067520     136     107  015     $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL ''G''
067525     015     012  123     $MSWR:  .ASCIZ  <15><12>/SWR - /
067536     040     040  116     $MNEW:  .ASCIZ  /  NEW = /
                                .EVEN
```

```
3 067550  012737  177777  001326  SHUT2:  MOV     #-1,C1LFG       ;SET THE CONTROL-C FLAG
4 067556  105737  001116                  TSTB    $TSTNM          ;DOING ANY TESTS ?
5 067562  001002                          BNE     1$              ;BR IF YES
6 067564  000137  037262                  JMP     SHUT            ;NO--RESPOND TO ^C
7 067570  000002                  1$:     RTI                     ;EXIT FROM INTERRUPT
```

```
                        .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                        ;;*****************************************************************
                        ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                        ;*CHANGE IT TO BINARY.
                        ;*CALL:
                        ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                        ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                        ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT

067572  011646         $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
067574  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
067602  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
067604  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
067606  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
067610  104412                1$:     RDLIN                   ;;READ AN ASCIZ LINE
067612  012600                        MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
067614  005001                        CLR     R1              ;;CLEAR DATA WORD
067616  005002                        CLR     R2
067620  112046                2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
067622  001412                        BEQ     3$              ;;IF ZERO GET OUT
067624  006301                        ASL     R1              ;;*2
067626  006102                        ROL     R2
067630  006301                        ASL     R1              ;;*4
067632  006102                        ROL     R2
067634  006301                        ASL     R1              ;;*8
067636  006102                        ROL     R2
067640  042716  177770                BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
067644  062601                        ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
067646  000764                        BR      2$              ;;LOOP
067650  005726                3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
067652  010166  000012                MOV     R1,12(SP)       ;;SAVE THE RESULT
067656  010237  067672                MOV     R2,$HIOCT
067662  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
067664  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
067666  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
067670  000002                        RTI                     ;;RETURN
067672  000000              $HIOCT:   .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                .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.

067674  016646  000002          $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
067700  042716  000020                  BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
067704  012746  067712                  MOV     #1$,-(SP)       ;; T-BIT TRAPS
067710  000002                          RTI                     ;;SET THE NEW STATUS
067712  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
067714  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
067720  005740                          TST     -(R0)           ;;BACKUP BY 2
067722  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
067724  006300                          ASL     R0              ;;POSITION FOR INDEXING
067726  016000  067746                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
067732  000200                          RTS     R0              ;;GO TO ROUTINE


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

067734  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
067736  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
067744  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
067746  067734                  $TRPAD: .WORD   $TRAP2
067750  063520                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
067752  063316                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
067754  063272                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
067756  063332                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
067760  063046                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
067762  062772                          $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

067764  066700                          $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING

067766  066610                          $CKSWR  ;;CALL=CKSWR    TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
067770  067152                          $RDCHR  ;;CALL=RDCHR    TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
067772  067242                          $RDLIN  ;;CALL=RDLIN    TRAP+12(104412) TTY TYPEIN STRING ROUTINE
067774  067572                          $RDOCT  ;;CALL=RDOCT    TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
067776  062676                          $SAVREG ;;CALL=SAVREG   TRAP+14(104414) SAVE R0-R5 ROUTINE
070000  062734                          $RESREG ;;CALL=RESREG   TRAP+15(104415) RESTORE R0-R5 ROUTINE
```

```
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
070002  012737  070142  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
070010  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
070016  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
070020  010146                        MOV     R1,-(SP)        ;;PUSH P1 ON STACK
070022  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
070024  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
070026  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
070030  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
070032  017746  111116                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
070036  010637  070146                MOV     SP,$SAVR6       ;;SAVE SP
070042  012737  070054  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
070050  000000                        HALT
070052  000776                        BR      .-2             ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
070054  012737  070142  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
070062  013706  070146                MOV     $SAVR6,SP       ;;GET SP
070066  005037  070146                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
070072  005237  070146         1$:    INC     $SAVR6          ;;WAIT FOR THE INC
070076  001375                        BNE     1$              ;;OF  WORD
070100  012677  111050                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
070104  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
070106  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
070110  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
070112  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
070114  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
070116  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
070120  012737  070002  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
070126  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
070134  104401                        TYPE                    ;REPORT THE POWER FAILURE
070136  070150           $PWRMG:      .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
070140  000002                        RTI
070142  000000           $ILLUP:      HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
070144  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
070146  000000           $SAVR6:      0                       ;;PUT THE SP HERE
070150     015     012     120  $POWER: .ASCIZ  <15><12>'POWER'
                                     .EVEN
```

```
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;********************************************************** *******
070160  112737  000001  070424  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
070166  112737  000001  070422  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
070174  000403                        BR      $ATYC
070176  112737  000001  070424  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
070204                          $ATYC:
070204  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
070206  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
070210  105737  070422                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
070214  001450                        BEQ     5$              ;;IF NOT:  BR
070216  122737  000001  001242        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
070224  001031                        BNE     3$              ;;IF NOT:  BR
070226  132737  000100  001243        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
070234  001425                        BEQ     3$              ;;IF NOT:  BR
070236  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
070242  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
070250  005737  001222        1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
070254  001375                        BNE     1$              ;;IF NOT:  WAIT
070256  010037  001236                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
070262  105720                2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
070264  001376                        BNE     2$
070266  163700  001236                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
070272  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
070274  010037  001240                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
070300  012737  000004  001222        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
070306  000413                        BR      5$
070310  017637  000004  070334  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
070316  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
070324  013746  177776                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
070330  004737  063520                JSR     PC,$TYPE        ;;CALL TYPE MACRO
070334  000000                4$:     .WORD   0
070336                          5$:
070336  105737  070424        10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
070342  001416                        BEQ     12$             ;;IF NOT:  BR
070344  005737  001242                TST     $ENV            ;;RUNNING UNDER APT?
070350  001413                        BEQ     12$             ;;IF NOT:  BR
070352  005737  001222        11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
070356  001375                        BNE     11$             ;;IF NOT:  WAIT
070360  017637  000004  001224        MOV     @4(SP),$FATAL   ;;GET ERROR #
070366  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
070374  005237  001222                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
070400  105037  070424        12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
070404  105037  070423                CLRB    $LFLG           ;;CLEAR LOG FLAG
070410  105037  070422                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
070414  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
070416  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
070420  000207                        RTS     PC              ;;RETURN
070422     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
070423     000          $LFLG:  .BYTE   0               ;;LOG FLAG
070424     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
                                .EVEN
            000200      APTSIZE = 200
            000001      APTENV  = 001
            000100      APTSPOOL= 100
            000040      APTCSUP = 040
```

```
                                        .SBTTL  CONSOLE MESSAGES
                                ;
000426  200     103     101     SCTMSG: .ASCIZ  <CRLF>@CANNOT RECOVER THE BAD SECTOR FILES ON THIS DEVICE@
000512  075     000             EQUALS: .ASCIZ  @=@
000514  101     114     114     ALL:    .ASCIZ  @ALL@<CRLF>
000521  040     077     040     QUES:   .ASCIZ  @ ? @
000525  054     040     000     COMMA:  .ASCIZ  @, @
000530  200     124     117     MSHELP: .ASCII  <CRLF>@TO  ENSURE THAT NO  BAD HEADERS ARE LEFT  ON THE DISK@
000616  200     120     101             .ASCII  <CRLF>@PACK,  THIS PROGRAM SHOULD BE HALTED BY TYPING A (^C)@
000704  200     103     117             .ASCII  <CRLF>@CONTROL C.  AS A RESULT,  THE PROGRAM WILL  BE HALTED@
000772  200     127     110             .ASCII  <CRLF>@WHEN THE DRIVE UNDER TEST HAS COMPLETED TESTING.@<CRLF>
001054  200     124     131             .ASCIZ  <CRLF>@TYPE HELP TEXT (L) N ? @
001105  200     122     115     CNSL01: .ASCIZ  <CRLF>@RMCS1=@
001115  040     114     111     CNSL02: .ASCIZ  @ LIMITS - LO= 160000, HI= 17XXXX@<CRLF>
001157  122     115     126     CNSL03: .ASCIZ  @RMVEC=@
001166  040     114     111     CNSL04: .ASCIZ  @ LIMITS - LO= 0, HI= 1000@<CRLF><LF>
001222  200     124     131     CNSL07: .ASCII  <CRLF>@TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S)@
001307  200     101     116             .ASCIZ  <CRLF>@AND TERMINATE INPUT WITH A CARRIAGE RETURN.@
001364  200             000     CNSL08: .ASCII  <CRLF>
001365  040     077     111     CNSL09: .ASCIZ  @ ?ILLEGAL INPUT@<CRLF>
001406  200     104     122     MSDRVS: .ASCIZ  <CRLF>/DRIVE(S): /
001422  104     122     111     MSGDRV: .ASCIZ  /DRIVE/
001430  200     125     116     SYSTAT: .ASCIZ  <CRLF>/UNIT STATUS:/
001446  122     115     060     $RM02:  .ASCIZ  /RM02/
001453  122     115     060     $RM03:  .ASCIZ  /RM03/
001460  122     115     060     $RM05:  .ASCIZ  /RM05/
001465  040     116     117     NOTRM:  .ASCIZ  @ NOT AN RM05/3/2@
001506  040     114     117     LODEV:  .ASCIZ  / LOAD DEVICE/
001523  040     116     117     NOTPRS: .ASCIZ  / NOT PRESENT/
001540  040     116     117     NOTAVL: .ASCIZ  / NOT AVAILABLE/
001557  040     117     106     UNTOFF: .ASCIZ  / OFFLINE/
001570  040     117     116     UNTON:  .ASCIZ  / ONLINE/
001600  200     104     122     DRIVES: .ASCIZ  <CRLF>/DRIVE(S) TO BE TESTED/
001627  116     117     116     NONE:   .ASCIZ  /NONE/
001634  116     000             N:      .ASCIZ  /N/
001636  131     000             Y:      .ASCIZ  /Y/
001640  040                     BLNKS4: .ASCII  / /
001641  040                     BLNKS3: .ASCII  / /
001642  040                     BLNKS2: .ASCII  / /
001643  040     000             BLNKS1: .ASCIZ  / /
                                .EVEN
```

```
                        .SBTTL  FUNCTION CODE TABLE

                        ;THE FUNCTION CODE TABLE IS USED TO DEFINE STATUS CONDITIONS FOR
                        ;EACH FUNCTION CODE.  BIT USAGE IS AS FOLLOWS:

                        ;       ATA - BIT 15 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF ATA SHOULD BE SET WHEN THE FUNCTION CODE IS EXECUTED, OTHERWISE,
                        ;BIT 15 IS ZERO, INDICATING THAT ATA SHOULD NOT NORMALLY BE SET.
                        ;NOTE THAT ATA MAY BE SET WHEN A COMMAND IS EXECUTED EVEN THOUGH
                        ;IT IS NOT EXPECTED AS A RESULT OF THE COMMAND.

                        ;       WCE - BIT 14 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF WRITE CHECK ERRORS ARE ENABLED AS A FUNCTION OF THE COMMAND.

                        ;       OPI - BIT 13 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF OPI ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.

                        ;       IVC - BIT 12 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF IVC ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.

                        ;       WLE - BIT 11 'S SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF WRITE ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.
                        ;THE WRITE ERRORS WHICH ARE ENABLED ARE 'WLE', 'WCF', 'DPE', 'UPE'.

                        ;       IAE - BIT 10 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF INVALID ADDRESS ERROR IS ENABLED FOR THAT COMMAND.

                        ;       AOE - BIT 09 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF READ AND WRITE ERRORS ARE ENABLED DURING THE EXECUTION OF THE
                        ;COMMAND.  THE ERRORS ENABLED BY THIS BIT ARE 'TRE', 'DLT', 'NEM',
                        ;'MXF', 'LBT', AND 'AOE'.

                        ;               BIT 08 IS NOT USED.

                        ;       HCE - BIT 07 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF HEADER ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.
                        ;HEADER ERRORS INCLUDE 'HCRC', 'HCE', 'FER', AND 'BSE'.

                        ;       ECH - BIT 06 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF DATA FIELD ERRORS ARE ENABLED DURING THE EXECUTION OF THAT
                        ;COMMAND.  THESE ERRORS INCLUDE 'MDPE', 'DCK', AND 'ECH'.

                        ;               BIT 05 IS NOT USED.

                        ;               BIT 04 IS NOT USED.

                        ;               BIT 03 IS NOT USED.

                        ;               BIT 02 IS NOT USED.

                        ;               BIT 01 IS NOT USED.

                        ;       ILF - BIT 00 IS SET IF THE FUNCTION CODE IS ILLEGAL.

071646                  FNCDTB:                                 ;FUNCTION CODE TABLE

071646  020000C         .WORD   OPI                             ;NOP
```

```
 58 071650  130001                .WORD   OPI!ATA.ILF.IVC                ;ILLEGAL FUNCTION (2)
 59 071652  132000                .WORD   ATA.OPI!IVC!IAE                 ;SEEK
 60 071654  130000                .WORD   ATA.OPI.IVC                     ;RECALIBRATE
 61 071656  020000                .WORD   OPI                             ;DRIVE CLEAR
 62 071660  030000                .WORD   OPI!IVC                         ;RELEASE
 63 071662  130000                .WORD   OPI.ATA.IVC                     ;OFFSET
 64 071664  130000                .WORD   OPI.ATA.IVC                     ;RETURN TO CENTERLINE
 65 071666  020000                .WORD   OPI                             ;READ IN PRESET
 66 071670  020000                .WORD   OPI                             ;PACK ACKNOWLEDGE
 67 071672  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (24)
 68 071674  130001                .WORD   OPI!ATA.ILF!IVC                 ;ILLEGAL FUNCTION (26)
 69 071676  132000                .WORD   ATA!OPI!IVC.IAE                 ;SEARCH
 70 071700  130001                .WORD   OPI!ATA!ILF.IVC                 ;ILLEGAL FUNCTION (32)
 71 071702  130001                .WORD   OPI.ATA.ILF!IVC                 ;ILLEGAL FUNCTION (34)
 72 071704  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (36)
 73 071706  130001                .WORD   OPI.ATA!ILF!IVC                 ;ILLEGAL FUNCTION (40)
 74 071710  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (42)
 75 071712  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (44)
 76 071714  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (46)
 77 071716  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
 78 071720  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE.ECH     ;WRITE CHECK HEADER AND DATA
 79 071722  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (54)
 80 071724  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (56)
 81 071726  037200                .WORD   OPI!IVC!WLE!IAE!AOE!HCE         ;WRITE DATA
 82 071730  037000                .WORD   OPI!IVC!WLE!IAE!AOE             ;WRITE HEADER AND DATA
 83 071732  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (64)
 84 071734  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (66)
 85 071736  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ DATA
 86 071740  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ HEADER AND DATA
 87 071742  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (74)
 88 071744  130001                .WORD   OPI.ATA!ILF!IVC                 ;ILLEGAL FUNCTION (76)
```

```
                        .SBTTL  ATTENTION (ATA) TABLE

                        ATNTBL: .BYTE   1.
                                .BYTE   2.
                                .BYTE   4.
                                .BYTE   8.
                                .BYTE   16.
                                .BYTE   32.
                                .BYTE   64.
                                .BYTE   128.
```

```
                                        .SBTTL  DATA PATTERN TABLE

                                RGDTPT:
                                MIXED:
071756                                          .WORD   0.
071756  000000                                  .WORD   0.
071760  000001                                  .WORD   1.
071762  000003                                  .WORD   3.
071764  000007                                  .WORD   7.
071766  000017                                  .WORD   15.
071770  000037                                  .WORD   31.
071772  000077                                  .WORD   63.
071774  000177                                  .WORD   127.
071776  000377                                  .WORD   255.
072000  000777                                  .WORD   511.
072002  001777                                  .WORD   1023.
072004  003777                                  .WORD   2047.
072006  007777                                  .WORD   4095.
072010  017777                                  .WORD   8191.
072012  037777                                  .WORD   16383.
072014  077777                                  .WORD   32767.
072016  177777                          ONES:   .WORD   65535.
072020  177777                                  .WORD   65535.
072022  077777                                  .WORD   32767.
072024  037777                                  .WORD   16383.
072026  017777                                  .WORD   8191.
072030  007777                                  .WORD   4095.
072032  003777                                  .WORD   2047.
072034  001777                                  .WORD   1023.
072036  000777                                  .WORD   511.
072040  000377                                  .WORD   255.
072042  000177                                  .WORD   127.
072044  000077                                  .WORD   63.
072046  000037                                  .WORD   31.
072050  000017                                  .WORD   15.
072052  000007                                  .WORD   7.
072054  000003                                  .WORD   3.
072056  000001                                  .WORD   1.
072060  000000                          ZEROS:  .WORD   0.
072062  000000                                  .WORD   0.
072064  000001                                  .WORD   1.
072066  000002                                  .WORD   2.
072070  000004                                  .WORD   4.
072072  000010                                  .WORD   8.
072074  000020                                  .WORD   16.
072076  000040                                  .WORD   32.
072100  000100                                  .WORD   64.
072102  000200                                  .WORD   128.
072104  000400                                  .WORD   256.
072106  001000                                  .WORD   512.
072110  002000                                  .WORD   1024.
072112  004000                                  .WORD   2048.
072114  010000                                  .WORD   4096.
072116  020000                                  .WORD   8192.
072120  040000                                  .WORD   16384.
072122  100000                                  .WORD   32768.
072124  100000                                  .WORD   32768.
072126  040000                                  .WORD   16384.
```

```
 58 072130  020000                        .WORD   8192.
 59 072132  010000                        .WORD   4096.
 60 072134  004000                        .WORD   2048.
 61 072136  002000                        .WORD   1024.
 62 072140  001000                        .WORD   512.
 63 072142  000400                        .WORD   256.
 64 072144  000200                        .WORD   128.
 65 072146  000100                        .WORD   64.
 66 072150  000040                        .WORD   32.
 67 072152  000020                        .WORD   16.
 68 072154  000010                        .WORD   8.
 69 072156  000004                        .WORD   4.
 70 072160  000002                        .WORD   2.
 71 072162  000001                        .WORD   1.
 72 072164  000000                        .WORD   0.
 73 072166  177777                        .WORD   65535.
 74 072170  177776                        .WORD   65534.
 75 072172  177774                        .WORD   65532.
 76 072174  177770                        .WORD   65528.
 77 072176  177760                        .WORD   65520.
 78 072200  177740                        .WORD   65504.
 79 072202  177700                        .WORD   65472.
 80 072204  177600                        .WORD   65408.
 81 072206  177400                        .WORD   65280.
 82 072210  177000                        .WORD   65024.
 83 072212  176000                        .WORD   64512.
 84 072214  174000                        .WORD   63488.
 85 072216  170000                        .WORD   61440.
 86 072220  160000                        .WORD   57344.
 87 072222  140000                        .WORD   49152.
 88 072224  100000                        .WORD   32768.
 89 072226  000000                        .WORD   0.
 90 072230  000000                        .WORD   0.
 91 072232  100000                        .WORD   32768.
 92 072234  140000                        .WORD   49152.
 93 072236  160000                        .WORD   57344.
 94 072240  170000                        .WORD   61440.
 95 072242  174000                        .WORD   63488.
 96 072244  176000                        .WORD   64512.
 97 072246  177000                        .WORD   65024.
 98 072250  177400                        .WORD   65280.
 99 072252  177600                        .WORD   65408.
100 072254  177700                        .WORD   65472.
101 072256  177740                        .WORD   65504.
102 072260  177760                        .WORD   65520.
103 072262  177770                        .WORD   65528.
104 072264  177774                        .WORD   65532.
105 072266  177776                        .WORD   65534.
106 072270  177777                        .WORD   65535.
107 072272  125252              EARLY:    .WORD   43690.
108 072274  152525                        .WORD   43690./2
109 072276  125252                        .WORD   43690.
110 072300  177777                        .WORD   65535.
111 072302  177776                        .WORD   65534.
112 072304  177775                        .WORD   65533.
113 072306  177773                        .WORD   65531.
114 072310  177767                        .WORD   65527.
```

```
115 072312  177757                    .WORD   65519.
116 072314  177737                    .WORD   65503.
117 072316  177677                    .WORD   65471.
118 072320  177577                    .WORD   65407.
119 072322  177377                    .WORD   65279.
120 072324  176777                    .WORD   65023.
121 072326  175777                    .WORD   64511.
122 072330  173777                    .WORD   63487.
123 072332  167777                    .WORD   61439.
124 072334  157777                    .WORD   57343.
125 072336  137777                    .WORD   49151.
126 072340  077777                    .WORD   32767.
127 072342  077777                    .WORD   32767.
128 072344  137777                    .WORD   49151.
129 072346  157777                    .WORD   57343.
130 072350  167777                    .WORD   61439.
131 072352  173777                    .WORD   63487.
132 072354  175777                    .WORD   64511.
133 072356  176777                    .WORD   65023.
134 072360  177377                    .WORD   65279.
135 072362  177577                    .WORD   65407.
136 072364  177677                    .WORD   65471.
137 072366  177737                    .WORD   65503.
138 072370  177757                    .WORD   65519.
139 072372  177767                    .WORD   65527.
140 072374  177773                    .WORD   65531.
141 072376  177775                    .WORD   65533.
142 072400  177776                    .WORD   65534.
143 072402  177777                    .WORD   65535.
144 072404                  ENRGDT:
```

```
                                        .SBTTL  ERROR MESSAGE TABLE
                                
072404  076772  000000          EMT1:   .WORD   EMS1,0
072410  077041  077056  000000  EMT2:   .WORD   EMS2,EMS3,0
072416  077041  077121  000000  EMT3:   .WORD   EMS2,EMS4,0
072424  077164  077214  000000  EMT4:   .WORD   EMS5,EMS6,0
072432  077164  077326  000000  EMT5:   .WORD   EMS5,EMS10,0
072440  103766  101205  000000  EMT6:   .WORD   EMS167,EMS64,0
072446  101753  104013  000000  EMT7:   .WORD   EMS110,EMS170,0
072454  077261  000000          EMT10:  .WORD   EMS7,0
072460  077326  000000          EMT11:  .WORD   EMS10,0
072464  077370  077401  000000  EMT12:  .WORD   EMS11,EMS12,0
072472  077442  077453  077464  EMT13:  .WORD   EMS13,EMS14,EMS15,EMS16,0
072504  077536  101205  000000  EMT14:  .WORD   EMS17,EMS64,0
072512  077370  077621  000000  EMT15:  .WORD   EMS11,EMS21,0
072520  077370  077644  077773  EMT16:  .WORD   EMS11,EMS22,EMS27,0
072530  077370  077660  100004  EMT17:  .WORD   EMS11,EMS23,EMS30,0
072540  077370  077706  100004  EMT20:  .WORD   EMS11,EMS24,EMS30,0
072550  077370  077735  077773  EMT21:  .WORD   EMS11,EMS25,EMS27,0
072560  077370  077752  077773  EMT22:  .WORD   EMS11,EMS26,EMS27,0
072570  077370  100014  100004  EMT23:  .WORD   EMS11,EMS31,EMS30,0
072600  077370  100043  100004  EMT24:  .WORD   EMS11,EMS32,EMS30,0
072610  077370  100072  100004  EMT25:  .WORD   EMS11,EMS33,EMS30,0
072620  077370  100120  100004  EMT26:  .WORD   EMS11,EMS34,EMS30,0
072630  077370  100171  100004  EMT27:  .WORD   EMS11,EMS35,EMS30,0
072640  077370  100220  100004  EMT30:  .WORD   EMS11,EMS36,EMS30,0
072650  077370  100247  100004  EMT31:  .WORD   EMS11,EMS37,EMS30,0
072660  077370  100275  100004  EMT32:  .WORD   EMS11,EMS40,EMS30,0
072670  077370  100324  100004  EMT33:  .WORD   EMS11,EMS41,EMS30,0
072700  077370  100352  100004  EMT34:  .WORD   EMS11,EMS42,EMS30,0
072710  077370  100401  100004  EMT35:  .WORD   EMS11,EMS43,EMS30,0
072720  077370  100430  100004  EMT36:  .WORD   EMS11,EMS44,EMS30,0
072730  077370  100503  100004  EMT37:  .WORD   EMS11,EMS45,EMS30,0
072740  101271  077576  000000  EMT40:  .WORD   EMS66,EMS20,0
072746  101457  103170  101465  EMT41:  .WORD   EMS75,EMS141,EMS76,0
072756  103261  103271  101413  EMT42:  .WORD   EMS144,EMS145,EMS72,EMS76,0
072770  100575  100711  101465  EMT43:  .WORD   EMS47,EMS53,EMS76,0
073000  101516  100711  101465  EMT44:  .WORD   EMS77,EMS53,EMS76,0
073010  101544  100711  101465  EMT45:  .WORD   EMS100,EMS53,EMS76,0
073020  101572  100711  101465  EMT46:  .WORD   EMS101,EMS53,EMS76,0
073030  101223  101205  000000  EMT47:  .WORD   EMS65,EMS64,0
073036  077442  077464  101157  EMT50:  .WORD   EMS13,EMS15,EMS63,0
073046  077370  100546  100004  EMT51:  .WORD   EMS11,EMS46,EMS30,EMS67,0
073060  100575  100711  101341  EMT52:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS140,EMS141,0
073076  100575  100711  101341  EMT53:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS141,EMS164,0
073114  100624  100711  101341  EMT54:  .WORD   EMS50,EMS53,EMS67,0
073124  103216  103233  100711  EMT55:  .WORD   EMS142,EMS143,EMS53,EMS67,0
073136  100653  101413  101341  EMT56:  .WORD   EMS51,EMS72,EMS67,EMS115,EMS50,EMS70,0
073154  103766  101423  101341  EMT57:  .WORD   EMS167,EMS73,EMS67,0
073164  100774  101341  102153  EMT60:  .WORD   EMS56,EMS67,EMS115,EMS150,EMS152,EMS70,0
073202  101400  100774  101341  EMT61:  .WORD   EMS71,EMS56,EMS67,EMS115,EMS150,EMS152,EMS72,0
073222  103147  101341  102153  EMT62:  .WORD   EMS140,EMS67,EMS115,EMS47,EMS70,0
073236  000000                  EMT63:  .WORD
073240  103354  103416  101366  EMT64:  .WORD   EMS150,EMS152,EMS70,EMS115,EMS56,EMS73,EMS67,0
073260  100701  104413  104574  EMT65:  .WORD   EMS52,EMS205,EMS214,EMS206,EMS115,EMS51,EMS72,0
073300  103725  101647  101413  EMT66:  .WORD   EMS165,EMS103,EMS72,EMS124,0
073312  103725  101647  101413  EMT67:  .WORD   EMS165,EMS103,EMS72,EMS171,0
```

```
 58 073324  100546  077576  103620  EMT70:  .WORD   EMS46,EMS20,EMS163,0
 59 073334  101400  101572  103620  EMT71:  .WORD   EMS71,EMS101,EMS163,0
 60 073344  100575  101413  103620  EMT72:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS140,EMS141,0
 61 073362  100575  101413  103620  EMT73:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS141,EMS72,0
 62 073400  100774  100711  103620  EMT74:  .WORD   EMS56,EMS53,EMS163,0
 63 073410  101400  100774  103620  EMT75:  .WORD   EMS71,EMS56,EMS163,EMS115,EMS150,EMS152,EMS72,0
 64 073430  100624  100711  103620  EMT76:  .WORD   EMS50,EMS53,EMS163,0
 65 073440  103216  103233  100711  EMT77:  .WORD   EMS142,EMS143,EMS53,EMS163,0
 66 073452  101457  103170  103620  EMT100: .WORD   EMS75,EMS141,EMS163,EMS115,EMS47,EMS70.0
 67 073470  101457  103354  103620  EMT101: .WORD   EMS75,EMS150,EMS163,EMS115,EMS56,EMS73,0
 68 073506  103766  101423  103620  EMT102: .WORD   EMS167,EMS73,EMS163,0
 69 073516  103337  103372  101440  EMT103: .WORD   EMS147,EMS151,EMS74,EMS163,0
 70 073530  100653  101440  103620  EMT104: .WORD   EMS51,EMS74,EMS163,EMS115,EMS50,EMS70,0
 71 073546  103725  101544  103620  EMT105: .WORD   EMS165,EMS100,EMS163,0
 72 073556  103725  101516  103620  EMT106: .WORD   EMS165,EMS77,EMS163,0
 73 073566  103261  103271  100711  EMT107: .WORD   EMS144,EMS145,EMS53,EMS163,EMS115,EMS143,EMS70,0
 74 073606  101753  102013  102156  EMT110: .WORD   EMS110,EMS112,EMS116,EMS111,0
 75 073620  102077  077121  000000  EMT111: .WORD   EMS113,EMS4,0
 76 073626  102077  077056  000000  EMT112: .WORD   EMS113,EMS3,0
 77 073634  101753  102153  102156  EMT113: .WORD   EMS110,EMS115,EMS116,EMS117,EMS114,0
 78 073650  102077  102230  000000  EMT114: .WORD   EMS113,EMS120,0
 79 073656  102251  102711  102735  EMT115: .WORD   EMS121,EMS132,EMS133,0
 80 073666  102314  102711  102735  EMT116: .WORD   EMS122,EMS132,EMS133,0
 81 073676  102351  102711  102735  EMT117: .WORD   EMS123,EMS132,EMS133,0
 82 073706  102414  102711  102735  EMT120: .WORD   EMS124,EMS132,EMS133,0
 83 073716  102446  102711  102735  EMT121: .WORD   EMS125,EMS132,EMS133,0
 84 073726  102511  102711  102735  EMT122: .WORD   EMS126,EMS132,EMS133,0
 85 073736  103112  102711  102735  EMT123: .WORD   EMS137,EMS132,EMS133,0
 86 073746  102613  102711  102735  EMT124: .WORD   EMS130,EMS132,EMS133,0
 87 073756  102651  102711  102735  EMT125: .WORD   EMS131,EMS132,EMS133,0
 88 073766  102251  102711  102760  EMT126: .WORD   EMS121,EMS132,EMS134,EMS123,0
 89 074000  102314  102711  102760  EMT127: .WORD   EMS122,EMS132,EMS134,EMS123,0
 90 074012  102351  102711  102760  EMT130: .WORD   EMS123,EMS132,EMS134,EMS123,0
 91 074024  102414  102711  102760  EMT131: .WORD   EMS124,EMS132,EMS134,EMS123 0
 92 074036  102446  102711  102760  EMT132: .WORD   EMS125,EMS132,EMS134,EMS123,0
 93 074050  102511  102711  102760  EMT133: .WORD   EMS126,EMS132,EMS134,EMS123,0
 94 074062  103112  102711  102760  EMT134: .WORD   EMS137,EMS132,EMS134,EMS123,0
 95 074074  102613  102711  102760  EMT135: .WORD   EMS130,EMS132,EMS134,EMS123,0
 96 074106  102651  102711  102760  EMT136: .WORD   EMS131,EMS132,EMS134,EMS123,0
 97 074120  102251  102711  103022  EMT137: .WORD   EMS121,EMS132,EMS135,0
 98 074130  102351  102711  103022  EMT140: .WORD   EMS123,EMS132,EMS135,0
 99 074140  102251  102711  103065  EMT141: .WORD   EMS121,EMS132,EMS136,0
100 074150  103112  102711  103065  EMT142: .WORD   EMS137,EMS132,EMS136,0
101 074160  102414  102711  103065  EMT143: .WORD   EMS124,EMS132,EMS136,0
102 074170  102446  102711  103065  EMT144: .WORD   EMS125,EMS132,EMS136,0
103 074200  102511  102711  103065  EMT145: .WORD   EMS126,EMS132,EMS136,0
104 074210  102651  102711  103065  EMT146: .WORD   EMS131,EMS132,EMS136,0
105 074220  104042  102711  103065  EMT147: .WORD   EMS171,EMS132,EMS136,0
106 074230  102613  102711  103065  EMT150: .WORD   EMS130,EMS132,EMS136,0
107 074240  103147  102153  103170  EMT151: .WORD   EMS140,EMS115,EMS141,EMS70,0
108 074252  103216  102153  103233  EMT152: .WORD   EMS142,EMS115,EMS143,EMS72,0
109 074264  103261  103271  103320  EMT153: .WORD   EMS144,EMS145,EMS146,EMS115,EMS143,EMS72,0
110 074302  103261  103271  077576  EMT154: .WORD   EMS144,EMS145,EMS20,EMS115,EMS143,EMS70,0
111 074320  103354  103416  103464  EMT155: .WORD   EMS150,EMS152,EMS154,EMS153,0
112 074332  103337  103372  103464  EMT156: .WORD   EMS147,EMS151,EMS154,EMS155,0
113 074344  103337  103372  103520  EMT157: .WORD   EMS147,EMS151,EMS156,EMS157,0
114 074356  103570  103520  103553  EMT160: .WORD   EMS161,EMS156,EMS160,0
```

```
115 074366  077735  077773  103520  EMT161: .WORD   EMS25,EMS27,EMS156,EMS160,0
116 074400  077752  077773  103520  EMT162: .WORD   EMS26,EMS27,EMS156,EMS160,0
117 074412  100575  103620  103147  EMT163: .WORD   EMS47,EMS163,EMS140,0
118 074422  100575  077576  000000  EMT164: .WORD   EMS47,EMS20,0
119 074430  100774  077576  000000  EMT165: .WORD   EMS56,EMS20,0
120 074436  100546  077576  000000  EMT166: .WORD   EMS46,EMS20,0
121 074444  105114  077576  000000  EMT167: .WORD   EMS224,EMS20,0
122 074452  104352  103170  100004  EMT170: .WORD   EMS203,EMS141,EMS30,EMS202,0
123 074464  103766  103464  103536  EMT171: .WORD   EMS167,EMS154,EMS157,0
124 074474  103766  103464  103500  EMT172: .WORD   EMS167,EMS154,EMS155,0
125 074504  104042  102711  102735  EMT173: .WORD   EMS171,EMS132,EMS133,0
126 074514  104042  102711  102760  EMT174: .WORD   EMS171,EMS132,EMS134,EMS123,0
127 074526  102077  104215  000000  EMT175: .WORD   EMS113,EMS177,0
128 074534  104237  104254  000000  EMT176: .WORD   EMS200,EMS201,0
129 074542  000000                  EMT177: .WORD
130 074544  104352  100653  100004  EMT200: .WORD   EMS203,EMS51,EMS30,EMS202,0
131 074556  104352  104365  100004  EMT201: .WORD   EMS203,EMS204,EMS30,EMS202,0
132 074570  104352  100624  100004  EMT202: .WORD   EMS203,EMS50,EMS30,EMS202,0
133 074602  104352  103233  100004  EMT203: .WORD   EMS203,EMS143,EMS30,EMS202,0
134 074614  103354  101440  104322  EMT204: .WORD   EMS150,EMS74,EMS202,EMS115,EMS152,EMS72,0
135 074632  100624  101647  101413  EMT205: .WORD   EMS50,EMS103,EMS72,0
136 074642  101656  101423  104322  EMT206: .WORD   EMS104,EMS73,EMS202,0
137 074652  101457  103170  102153  EMT207: .WORD   EMS75,EMS141,EMS115,EMS140,0
138 074664  101457  103354  000000  EMT210: .WORD   EMS75,EMS150,0
139 074672  100653  101413  102153  EMT211: .WORD   EMS51,EMS72,EMS115,EMS50,EMS70,0
140 074706  103216  100711  102153  EMT212: .WORD   EMS142,EMS53,EMS115,EMS143,EMS72,0
141 074722  100624  101647  100711  EMT213: .WORD   EMS50,EMS103,EMS53,0
142 074732  100701  104413  103741  EMT214: .WORD   EMS52,EMS205,EMS166,EMS206,EMS115,EMS51,EMS72,0
143 074752  100701  102206  100774  EMT215: .WORD   EMS52,EMS117,EMS56,EMS163,0
144 074764  100774  100004  101205  EMT216: .WORD   EMS56,EMS30,EMS64,0
145 074774  100546  100004  101205  EMT217: .WORD   EMS46,EMS30,EMS64,0
146 075004  100014  100004  101205  EMT220: .WORD   EMS31,EMS30,EMS64,0
147 075014  100575  100004  101205  EMT221: .WORD   EMS47,EMS30,EMS64,0
148 075024  100701  102206  101516  EMT222: .WORD   EMS52,EMS117,EMS77,0
149 075034  100701  102206  101157  EMT223: .WORD   EMS52,EMS117,EMS63,0
150 075044  000000                  EMT224: .WORD
151 075046  000000                  EMT225: .WORD
152 075050  000000                  EMT226: .WORD
153 075052  000000                  EMT227: .WORD
154 075054  000000                  EMT230: .WORD
155 075056  000000                  EMT231: .WORD
156 075060  000000                  EMT232: .WORD
157 075062  000000                  EMT233: .WORD
158 075064  000000                  EMT234: .WORD
159 075066  000000                  EMT235: .WORD
160 075070  000000                  EMT236: .WORD
161 075072  000000                  EMT237: .WORD
162 075074  000000                  EMT240: .WORD
163 075076  000000                  EMT241: .WORD
164 075100  000000                  EMT242: .WORD
165 075102  000000                  EMT243: .WORD
166 075104  000000                  EMT244: .WORD
167 075106  000000                  EMT245: .WORD
168 075110  103766  102711  104452  EMT246: .WORD   EMS167,EMS132,EMS207,0
169 075120  103766  102711  104477  EMT247: .WORD   EMS167,EMS132,EMS210,EMS125,0
170 075132  103766  104510  104477  EMT250: .WORD   EMS167,EMS211,EMS210,EMS207,EMS206,0
171 075146  104526  104551  000000  EMT251: .WORD   EMS212,EMS213,0
```

```
172 075154  103725  104526  000000  EMT252: .WORD   EMS165,EMS212,0
173 075162  103337  103372  103520  EMT253: .WORD   EMS147,EMS151,EMS156,EMS210,EMS26,EMS27,0
174 075200  104352  101572  100004  EMT254: .WORD   EMS203,EMS101,EMS30,0
175 075210  104352  103766  100004  EMT255: .WORD   EMS203,EMS167,EMS30,0
176 075220  104352  101544  100004  EMT256: .WORD   EMS203,EMS100,EMS30,0
177 075230  077370  100546  100004  EMT257: .WORD   EMS11,EMS46,EMS30,EMS102,0
178 075242  100701  102206  100774  EMT260: .WORD   EMS52,EMS117,EMS56,EMS102,0
179 075254  100701  104413  104716  EMT261: .WORD   EMS52,EMS205,EMS220,EMS206,EMS115,EMS51,EMS72,0
180 075274  101656  101423  104322  EMT262: .WORD   EMS104,EMS73,EMS202,0
181 075304  100624  100711  101620  EMT263: .WORD   EMS50,EMS53,EMS102,0
182 075314  100774  100711  101620  EMT264: .WORD   EMS56,EMS53,EMS102,EMS115,EMS150,EMS152,EMS70,0
183 075334  101400  100774  101620  EMT265: .WORD   EMS71,EMS56,EMS102,EMS115,EMS150,EMS152,EMS72,0
184 075354  103216  103233  100711  EMT266: .WORD   EMS142,EMS143,EMS53,EMS102,0
185 075366  100624  103320  102153  EMT267: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,0
186 075404  100575  100711  101620  EMT270: .WORD   EMS47,EMS53,EMS102,EMS115,EMS140,0
187 075420  100575  100711  101620  EMT271: .WORD   EMS47,EMS53,EMS102,EMS115,EMS141,EMS72,0
188 075436  101457  103170  101620  EMT272: .WORD   EMS75,EMS141,EMS102,EMS115,EMS47,EMS73,0
189 075454  100653  101440  101620  EMT273: .WORD   EMS51,EMS74,EMS102,EMS115,EMS50,EMS70,0
190 075472  101157  100711  101051  EMT274: .WORD   EMS63,EMS53,EMS57,EMS115,EMS41,EMS146,0
191 075510  102156  100711  100324  EMT275: .WORD   EMS116,EMS53,EMS41,EMS57,0
192 075522  100575  100711  101051  EMT276: .WORD   EMS47,EMS53,EMS57,EMS115,EMS140,0
193 075536  100575  100711  101051  EMT277: .WORD   EMS47,EMS53,EMS57,EMS115,EMS141,EMS72,0
194 075554  100774  100711  101051  EMT300: .WORD   EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS70,0
195 075574  101400  100774  100711  EMT301: .WORD   EMS71,EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS72,0
196 075616  103725  101544  101647  EMT302: .WORD   EMS165,EMS100,EMS103,EMS57,0
197 075630  103725  101572  101647  EMT303: .WORD   EMS165,EMS101,EMS103,EMS57,0
198 075642  103725  101516  101647  EMT304: .WORD   EMS165,EMS77,EMS103,EMS57,0
199 075654  100546  100004  101205  EMT305: .WORD   EMS46,EMS30,EMS64,EMS57,0
200 075666  100624  100711  101051  EMT306: .WORD   EMS50,EMS53,EMS57,0
201 075676  100624  103320  102153  EMT307: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,EMS57,0
202 075716  103216  103233  100711  EMT310: .WORD   EMS142,EMS143,EMS53,EMS57,0
203 075730  101656  101647  100711  EMT311: .WORD   EMS104,EMS103,EMS53,EMS57,0
204 075742  101705  101647  100711  EMT312: .WORD   EMS105,EMS103,EMS53,EMS57,0
205 075754  103261  103271  101647  EMT313: .WORD   EMS144,EMS145,EMS103,EMS57,EMS115,EMS143,EMS70,0
206 075774  100352  101647  100711  EMT314: .WORD   EMS42,EMS103,EMS53,EMS57,0
207 076006  100014  101647  100711  EMT315: .WORD   EMS31,EMS103,EMS53,EMS57,0
208 076020  101400  100014  101647  EMT316: .WORD   EMS71,EMS31,EMS103,EMS57,0
209 076032  100401  101647  101051  EMT317: .WORD   EMS43,EMS103,EMS57,0
210 076042  100503  101647  101051  EMT320: .WORD   EMS45,EMS103,EMS57,0
211 076052  100430  101647  101051  EMT321: .WORD   EMS44,EMS103,EMS57,0
212 076062  101734  077576  000000  EMT322: .WORD   EMS106,EMS20,0
213 076070  100220  101647  101051  EMT323: .WORD   EMS36,EMS103,EMS57,0
214 076100  104115  100220  101647  EMT324: .WORD   EMS173,EMS36,EMS103,EMS57,0
215 076112  104075  100220  101647  EMT325: .WORD   EMS172,EMS36,EMS103,EMS57,0
216 076124  077442  104132  077464  EMT326: .WORD   EMS13,EMS174,EMS15,EMS35,EMS53,EMS175,0
217 076142  103337  103372  101440  EMT327: .WORD   EMS147,EMS151,EMS74,EMS175,0
218 076154  101271  100711  104140  EMT330: .WORD   EMS66,EMS53,EMS175,0
219 076164  100072  101647  100711  EMT331: .WORD   EMS33,EMS103,EMS53,EMS175,0
220 076176  100275  101647  100711  EMT332: .WORD   EMS40,EMS103,EMS53,EMS57,0
221 076210  100653  101440  101051  EMT333: .WORD   EMS51,EMS74,EMS57,EMS115,EMS50,EMS70,0
222 076226  101457  103170  101051  EMT334: .WORD   EMS75,EMS141,EMS57,EMS115,EMS47,EMS73,0
223 076244  101457  103354  103416  EMT335: .WORD   EMS75,EMS150,EMS152,EMS57,EMS115,EMS56,EMS73,0
224 076264  101077  101112  101141  EMT336: .WORD   EMS60,EMS61,EMS62,0
225 076274  102156  102206  100120  EMT337: .WORD   EMS116,EMS117,EMS34,0
226 076304  100120  100711  100723  EMT340: .WORD   EMS34,EMS53,EMS54,EMS111,0
227 076316  100745  100120  000000  EMT341: .WORD   EMS55,EMS34,0
228 076324  100701  102206  100774  EMT342: .WORD   EMS52,EMS117,EMS56,EMS57,0
```

```
 229 076336  100701  102206  100503  EMT343: .WORD   EMS52,EMS117,EMS45,EMS57,0
 230 076350  100701  102206  100430  EMT344: .WORD   EMS52,EMS117,EMS44,EMS57,0
 231 076362  100701  102206  104736  EMT345: .WORD   EMS52,EMS117,EMS221,0
 232 076372  101734  077576  102153  EMT346: .WORD   EMS106,EMS20,EMS115,EMS223,EMS72,0
 233 076406  100701  104413  105006  EMT347: .WORD   EMS52,EMS205,EMS222,EMS206,0
 234 076420  101457  103354  101620  EMT350: .WORD   EMS75,EMS150,EMS102,EMS115,EMS56,EMS73,0
 235 076436  103766  101423  101620  EMT351: .WORD   EMS167,EMS73,EMS102,0
 236 076446  104612  000000          EMT352: .WORD   EMS215,0
 237 076452  104663  104352  104633  EMT353: .WORD   EMS217,EMS203,EMS216,0
 238 076462  104736  077576  000000  EMT354: .WORD   EMS221,EMS20,0
```

```
 1 076470  105166  105772  106047  EHT1:    .WORD   EH1,STSH1,STSH2,STSH4,0
 2 076502  105772  106047  106174  EHT2:    .WORD   STSH1,STSH2,STSH4,0
 3
 4 076512  105205  000000          EHT110:  .WORD   EH110,0
 5 076516  105214  000000          EHT111:  .WORD   EH111,0
 6
 7 076522  105233  000000          EHT114:  .WORD   EH114,0
 8 076526  105262  105772  106047  EHT223:  .WORD   EH223,STSH1,STSH2,STSH4,0
 9 076540  105310  105772  106047  EHT256:  .WORD   EH256,STSH1,STSH2,STSH4,0
10
11 076552  105364  105772  106047  EHT336:  .WORD   EH336,STSH1,STSH2,STSH4,0
12 076564  105423  105772  106047  EHT337:  .WORD   EH337,STSH1,STSH2,STSH4,0
13 076576  105560  105772  106047  EHT344:  .WORD   EH344,STSH1,STSH2,STSH4,0
14
15 076610  105716  000000          EHT353:  .WORD   EH353,0
```

ERROR MESSAGE TABLE

```
 1 076614  106232  106326  106344  EDT1:    .WORD   ED1,STSD1,STSD2,STSD4
 2 076624  106326  106344  106376  EDT2:    .WORD   STSD1,STSD2,STSD4
 3
 4 076632  106240                  EDT110:  .WORD   ED110
 5 076634  106244                  EDT111:  .WORD   ED111
 6
 7 076636  106252                  EDT114:  .WORD   ED114
 8 076640  106262  106326  106344  EDT223:  .WORD   ED223,STSD1,STSD2,STSD4
 9
10 076650  106272  106326  106344  EDT336:  .WORD   ED336,STSD1,STSD2,STSD4
11 076660  106304  106326  106344  EDT337:  .WORD   ED337,STSD1,STSD2,STSD4
12 076670  106304  106326  106344  EDT344:  .WORD   ED337,STSD1,STSD2,STSD4,0
13
14 076702  106316                  EDT353:  .WORD   ED353
```

SEQ 0292

```
 1 076704  106411  106427  106427  EFT1:    .WORD   EF111,STSF,STSF,STSF
 2 076714  106427  106427  106427  EFT2:    .WORD   STSF,STSF,STSF
 3
 4 076722  106410                  EFT110:  .WORD   EF110
 5 076724  106411                  EFT111:  .WORD   EF111
 6
 7 076726  106413                  EFT114:  .WORD   EF114
 8 076730  106413  106427  106427  EFT223:  .WORD   EF114,STSF,STSF,STSF
 9
10 076740  106416  106427  106427  EFT336:  .WORD   EF336,STSF,STSF,STSF
11 076750  106416  106427  106427  EFT337:  .WORD   EF336,STSF,STSF,STSF
12 076760  106416  106427  106427  EFT344:  .WORD   EF336,STSF,STSF,STSF
13
14 076770  106413                  EFT353:  .WORD   EF114
```

```
                                .SBTTL  ERROR MESSAGE STRINGS

076772  127  122  117  EMS1:   .ASCIZ  @WRONG UNIT SELECTED (RMCS2, BITS 0-2) @
077041  104  105  126  EMS2:   .ASCIZ  @DEVICE WENT @
077056  125  116  101  EMS3:   .ASCIZ  @UNAVAILABLE 'DVA' (RMCS1, BIT 11) @
077121  116  117  116  EMS4:   .ASCIZ  @NONEXISTENT 'NED' (RMCS2, BIT 12) @
077164  103  117  115  EMS5:   .ASCIZ  @COMMAND NOT COMPLETED, @
077214  103  117  116  EMS6:   .ASCIZ  @CONTROLLER NOT READY (RMCS1, BIT 7) @
077261  104  122  111  EMS7:   .ASCIZ  @DRIVE NOT READY 'DRY' (RMDS, BIT 7) @
077326  107  117  040  EMS10:  .ASCIZ  @GO NOT RESET 'GO' (RMCS1, BIT 0) @
077370  111  116  126  EMS11:  .ASCIZ  @INVALID @
077401  106  125  116  EMS12:  .ASCIZ  @FUNCTION CODE (RMCS1, BITS 1-5) @
077442  115  101  123  EMS13:  .ASCIZ  @MASSBUS @
077453  103  117  116  EMS14:  .ASCIZ  @CONTROL @
077464  102  125  123  EMS15:  .ASCIZ  @BUS PARITY ERROR @
077506  042  115  103  EMS16:  .ASCIZ  @'MCPE' (RMCS1, BIT 13) @
077536  124  122  101  EMS17:  .ASCIZ  @TRANSFER ERROR (RMCS1, BIT 14) @
077576  123  110  117  EMS20:  .ASCIZ  @SHOULD NOT BE SET @
077621  127  117  122  EMS21:  .ASCIZ  @WORD COUNT (RMWC) @
077644  102  125  123  EMS22:  .ASCIZ  @BUS (RMBA) @
077660  042  114  102  EMS23:  .ASCIZ  @'LBT' (RMDS, BIT 10) @
077706  042  101  117  EMS24:  .ASCIZ  @'AOE' (RMER1, BIT 09) @
077735  104  111  123  EMS25:  .ASCIZ  @DISK (RMDA) @
077752  103  131  114  EMS26:  .ASCIZ  @CYLINDER (RMDC) @
077773  101  104  104  EMS27:  .ASCIZ  @ADDRESS @
100004  123  124  101  EMS30:  .ASCIZ  @STATUS @
100014  042  127  114  EMS31:  .ASCIZ  @'WLE' (RMER1, BIT 11) @
100043  042  125  120  EMS32:  .ASCIZ  @'UPE' (RMCS2, BIT 13) @
100072  042  127  103  EMS33:  .ASCIZ  @'WCF' (RMER1, BIT 5) @
100120  127  122  111  EMS34:  .ASCIZ  @WRITE CHECK ERROR-'WCE' (RMCS2, BIT 14) @
100171  042  115  104  EMS35:  .ASCIZ  @'MDPE' (RMCS2, BIT 8) @
100220  042  104  103  EMS36:  .ASCIZ  @'DCK' (RMER1, BIT 15) @
100247  042  105  103  EMS37:  .ASCIZ  @'ECH' (RMER1, BIT 6) @
100275  042  104  114  EMS40:  .ASCIZ  @'DLT' (RMCS2, BIT 15) @
100324  042  115  130  EMS41:  .ASCIZ  @'MXF' (RMCS2, BIT 9) @'
100352  042  104  124  EMS42:  .ASCIZ  @'DTE' (RMER1, BIT 12) @
100401  042  110  103  EMS43:  .ASCIZ  @'HCRC' (RMER1, BIT 8) @
100430  110  105  101  EMS44:  .ASCIZ  @HEADER COMPARE ERROR 'HCE' (RMER1, BIT 7) @
100503  106  117  122  EMS45:  .ASCIZ  @FORMAT ERROR 'FER' (RMER1, BIT 4) @
100546  042  111  101  EMS46:  .ASCIZ  @'IAE' (RMER1, BIT 10) @
100575  042  117  120  EMS47:  .ASCIZ  @'OPI' (RMER1, BIT 13) @
100624  042  123  113  EMS50:  .ASCIZ  @'SKI' (RMER2, BIT 14) @
100653  042  120  111  EMS51:  .ASCIZ  @'PIP' (RMDS, BIT 13) @
100701  124  110  105  EMS52:  .ASCIZ  @THE RM @
100711  104  105  124  EMS53:  .ASCIZ  @DETECTED @
100723  101  124  040  EMS54:  .ASCIZ  @AT AN UNEXPECTED @
100745  111  116  103  EMS55:  .ASCIZ  @INCORRECT DATA DURING @
100774  111  116  126  EMS56:  .ASCIZ  @INVALID COMMAND ERROR 'IVC' (RMER2, BIT 12) @
101051  104  125  122  EMS57:  .ASCIZ  @DURING DATA TRANSFER @
101077  104  101  124  EMS60:  .ASCIZ  @DATA READ @
101112  104  117  105  EMS61:  .ASCIZ  @DOES NOT COMPARE WITH @
101141  104  101  124  EMS62:  .ASCIZ  @DATA WRITTEN @
101157  042  120  101  EMS63:  .ASCIZ  @'PAR' (RMER1, BIT 3) @
101205  111  123  040  EMS64:  .ASCIZ  @IS INCORRECT @
101223  103  117  115  EMS65:  .ASCIZ  @COMPOSITE ERROR 'ERR' (RMDS, BIT 14) @
101271  104  101  124  EMS66:  .ASCIZ  @DATA PARITY ERROR 'DPE' (RMER2, BIT 3) @
101341  104  125  122  EMS67:  .ASCIZ  @DURING SEEK COMMAND @
```

ERROR MESSAGE STRINGS

```
 58 101366  111  123  040  EMS70:  .ASCIZ  @IS RESET @
 59 101400  105  122  122  EMS71:  .ASCIZ  @ERRONEOUS @
 60 101413  111  123  040  EMS72:  .ASCIZ  @IS SET @
 61 101423  104  111  104  EMS73:  .ASCIZ  @DID NOT SET @
 62 101440  104  111  104  EMS74:  .ASCIZ  @DID NOT RESET @
 63 101457  114  117  123  EMS75:  .ASCIZ  @LOST @
 64 101465  104  125  122  EMS76:  .ASCIZ  @DURING PACK ACK COMMAND @
 65 101516  042  122  115  EMS77:  .ASCIZ  @'RMR'' (RMER1, BIT 2) @
 66 101544  042  111  114  EMS100: .ASCIZ  @''ILR'' (RMER1, BIT 1) @
 67 101572  042  111  114  EMS101: .ASCIZ  @''ILF'' (RMER1, BIT 0) @
 68 101620  104  125  122  EMS102: .ASCIZ  @DURING SEARCH COMMAND @
 69 101647  105  122  122  EMS103: .ASCIZ  @ERROR @
 70 101656  042  114  102  EMS104: .ASCIZ  @'LBC'' (RMER2, BIT 10) @
 71 101705  042  114  123  EMS105: .ASCIZ  @'LSC'' (RMER2, BIT 11) @
 72 101734  110  105  101  EMS106: .ASCIZ  @HEADER ERRORS @
 73 101753  102  125  123  EMS110: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
 74 102002  101  104  104  EMS111: .ASCIZ  @ADDRESS @
 75 102013  127  110  105  EMS112: .ASCII  @WHEN READING/WRITING RH REGISTERS @
 76 102055  101  124  040          .ASCIZ  @AT THE FOLLOWING @
 77 102077  124  110  105  EMS113: .ASCIZ  @THE SELECTED DEVICE IS @
 78 102127  116  117  116  EMS114: .ASCIZ  @NONEXISTENT DEVICE @
 79 102153  015  012  000  EMS115: .ASCIZ  <CR><LF>
 80 102156  124  110  105  EMS116: .ASCIZ  @THE MASSBUS CONTROLLER @
 81 102206  106  101  111  EMS117: .ASCIZ  @FAILED TO DETECT @
 82 102230  116  117  124  EMS120: .ASCIZ  @NOT AN RM05/3/2 @
 83 102251  103  117  116  EMS121: .ASCIZ  @CONTROL STATUS REGISTER 1, RMCS1, @
 84 102314  102  125  123  EMS122: .ASCIZ  @BUS ADDRESS REGISTER, RMBA, @
 85 102351  103  117  116  EMS123: .ASCIZ  @CONTROL STATUS REGISTER 2, RMCS2, @
 86 102414  105  122  122  EMS124: .ASCIZ  @ERROR REGISTER 1, RMER1, @
 87 102446  101  124  124  EMS125: .ASCIZ  @ATTENTION SUMMARY REGISTER, RMAS, @
 88 102511  115  101  111  EMS126: .ASCIZ  @MAINTENANCE REGISTER #1, RMMR #1, @
 89 102554  105  103  103  EMS127: .ASCIZ  @ECC POSITION REGISTER, RMEC1, @
 90 102613  105  103  103  EMS130: .ASCIZ  @ECC PATTERN REGISTER, RMEC2, @
 91 102651  115  101  111  EMS131: .ASCIZ  @MAINTENANCE REGISTER 2, RMMR2, @
 92 102711  116  117  124  EMS132: .ASCIZ  @NOT INITIALIZED BY @
 93 102735  125  116  111  EMS133: .ASCIZ  @UNIBUS INITIALIZE @
 94 102760  103  117  116  EMS134: .ASCIZ  @CONTROLLER CLEAR, I.E.  BIT 5 OF @
 95 103022  122  110  057  EMS135: .ASCIZ  @RH/RM ERROR CLEAR (RMCS1, BIT 14) @
 96 103065  104  122  111  EMS136: .ASCIZ  @DRIVE CLEAR COMMAND @
 97 103112  104  122  111  EMS137: .ASCIZ  @DRIVE STATUS REGISTER, RMDS @
 98 103147  115  105  104  EMS140: .ASCIZ  @MEDIUM OFF LINE @
 99 103170  042  115  117  EMS141: .ASCIZ  @'MOL'' (RMDS, BIT 12) @
100 103216  104  122  111  EMS142: .ASCIZ  @DRIVE FAULT @
101 103233  042  104  126  EMS143: .ASCIZ  @'DVC'' (RMER2, BIT 7) @
102 103261  125  116  123  EMS144: .ASCIZ  @UNSAFE @
103 103271  042  125  116  EMS145: .ASCIZ  @'UNS'' (RMER1, BIT 14) @
104 103320  123  110  117  EMS146: .ASCIZ  @SHOULD BE SET @
105 103337  117  106  106  EMS147: .ASCIZ  @OFFSET MODE @
106 103354  126  117  114  EMS150: .ASCIZ  @VOLUME VALID @
107 103372  042  117  115  EMS151: .ASCIZ  @'OM'' (RMDS, BIT 0) @
108 103416  042  126  126  EMS152: .ASCIZ  @'VV'' (RMDS, BIT 6) @
109 103442  120  101  103  EMS153: .ASCIZ  @PACK ACK COMMAND @
110 103464  116  117  124  EMS154: .ASCIZ  @NOT SET BY @
111 103500  117  106  106  EMS155: .ASCIZ  @OFFSET COMMAND @
112 103520  116  117  124  EMS156: .ASCIZ  @NOT RESET BY @
113 103536  122  124  103  EMS157: .ASCIZ  @RTC COMMAND @
114 103553  122  111  120  EMS160: .ASCIZ  @RIP COMMAND @
```

```
115 103570      117     106     106  EMS161: .ASCIZ  @OFFSET REGISTER (RMOF) @
116 103620                           EMS162:         ;<UNUSED>
117 103620      104     125     122  EMS163: .ASCIZ  @DURING RECALIBRATE @
118 103644      111     123     040  EMS164: .ASCII  @IS INTERMITTENT OR DRIVE DIDNT DROP ON @
119 103713      103     131     114          .ASCIZ  @CYLINDER @
120 103725      125     116     105  EMS165: .ASCIZ  @UNEXPECTED @
121 103741      122     105     103  EMS166: .ASCIZ  @RECALIBRATE COMMAND @
122 103766      042     101     124  EMS167: .ASCIZ  @''ATA'' (RMDS, BIT15) @
123 104013      127     110     105  EMS170: .ASCIZ  @WHEN READING REGISTER @
124 104042      105     122     122  EMS171: .ASCIZ  @ERROR REGISTER #2, RMER2, @
125 104075      116     117     116  EMS172: .ASCIZ  @NONRECOVERABLE @
126 104115      122     105     103  EMS173: .ASCIZ  @RECOVERABLE @
127 104132      104     101     124  EMS174: .ASCIZ  @DATA @
128 104140      104     125     122  EMS175: .ASCIZ  @DURING WRITE COMMAND @
129 104166      042     117     120  EMS176: .ASCIZ  @''OPE'' (RMER2, BIT 13) @
130 104215      111     116     040  EMS177: .ASCIZ  @IN WRITE PROTECT @
131 104237      103     101     116  EMS200: .ASCIZ  @CAN NOT SET @
132 104254      104     111     101  EMS201: .ASCIZ  @DIAGNOSTIC MODE ''DMD'' (RMMR1, BIT 0) @
133 104322      104     125     122  EMS202: .ASCIZ  @DURING DIAGNOSTIC MODE @
134 104352      111     116     103  EMS203: .ASCIZ  @INCORRECT @
135 104365      042     127     122  EMS204: .ASCIZ  @''WRL'' (RMDS, BIT 11) @
136 104413      105     130     105  EMS205: .ASCIZ  @EXECUTED @
137 104425      127     111     124  EMS206: .ASCIZ  @WITH COMP ERROR SET @
138 104452      042     107     117  EMS207: .ASCIZ  @''GO'' (RMCS1, BIT 0) @
139 104477      127     122     111  EMS210: .ASCIZ  @WRITING @
140 104510      127     101     123  EMS211: .ASCIZ  @WAS RESET BY @
141 104526      120     122     117  EMS212: .ASCIZ  @PROGRAM INTERRUPT @
142 104551      127     101     123  EMS213: .ASCIZ  @WAS NOT GENERATED @
143 104574      123     105     105  EMS214: .ASCIZ  @SEEK COMMAND @
144 104612      120     122     117  EMS215: .ASCIZ  @PROGRAM TIMEOUT @
145 104633      104     125     122  EMS216: .ASCIZ  @DURING LOOK AHEAD TEST @
146 104663      114     117     117  EMS217: .ASCIZ  @LOOK AHEAD REGISTER,RMLA, @
147 104716      123     105     101  EMS220: .ASCIZ  @SEARCH COMMAND @
148 104736      102     101     104  EMS221: .ASCIZ  @BAD SECTOR ERROR ''BSE'' (RMER2, BIT 15) @
149 105006      101     040     104  EMS222: .ASCIZ  @A DATA TRANSFER COMMAND @
150 105037      110     105     101  EMS223: .ASCIZ  @HEADER COMPARE INHIBIT ''HCI'' (RMOF, BIT 10) @
151 105114      116     117     116  EMS224: .ASCIZ  @NONEXISTENT MEMORY ''NEM'' (RMCS2, BIT 11) @
```

```
     1 105166  105  130  120  EH1:    .ASCIZ  @EXPCTD  RECEVD@
     2 105205  102  125  123  EH110:  .ASCIZ  @BUSADR@
     3 105214  040  122  115  EH111:  .ASCIZ  @ RMCS2   RMCS1@
     4
     5 105233  122  105  103  EH114:  .ASCIZ  @RECEVD  SNGPRT  DULPRT@
     6 105262  105  130  120  EH223:  .ASCIZ  @EXPCTD  RECEVD   DATA@
     7 105310  105  130  120  EH256:  .ASCII  @EXPCTD  RECEVD  RGSTR@<CRLF>
     8 105336  123  124  101          .ASCIZ  @STATUS  STATUS  INDEX@
     9
    10 105364  107  104  101  EH336:  .ASCIZ  @GDADRS  GDDATA  BDADRS  BDDATA@
    11 105423  122  115  103  EH337:  .ASCII  @RMCS2   STATUS  FAILING   DATA@<CRLF>
    12 105462  137  137  137          .ASCII  @                          @<CRLF>
    13 105521  105  130  120          .ASCIZ  @EXPCTD--RECEVD  --BIT---ADRESS@
    14
    15 105560  122  115  105  EH344:  .ASCII  @RMER1   STATUS  HEADER  FAILING@<CRLF>
    16 105620  137  137  137          .ASCII  @                 WORD     BIT@<CRLF>
    17 105656  105  130  120          .ASCIZ  @EXPCTD--RECEVD  NUMBER  POSITON@
    18 105716  105  130  120  EH353:  .ASCII  @EXPCTD  RECEVD@<CRLF>
    19 105735  074  103  122          .ASCIZ  @<CRLF> RMLA    RMLA    RMOF @
    20
    21 105772  040  122  115  STSH1:  .ASCII  @ RMCS1  RMCS2   RMDS    RMER1   RMER2@
    22 106037  040  040  040          .ASCIZ  @   RMAS@
    23 106047  040  122  115  STSH2:  .ASCII  @ RMWC    RMBA    RMDA    RMOF    RMDC@
    24 106114  040  040  040          .ASCIZ  @    RMEC1   RMEC2@
    25 106136  040  122  115  STSH3:  .ASCIZ  @ RMDA    RMDC    RMOF    RMLA@
    26 106174  040  122  115  STSH4:  .ASCIZ  @ RMMR1   RMMR2   RMDT    RMSN@
    27                               .EVEN
```

ERROR MESSAGE STRINGS

```
      1 106232  001140  001142  000000  ED1:    .WORD   $GDDAT,$BDDAT,0
      2 106240  001276  000000          ED110:  .WORD   $BASE,0
      3 106244  001174  001176  000000  ED111:  .WORD   $TMP0,$TMP1,0
      4
      5 106252  001364  001176  001200  ED114:  .WORD   RMDTI,$TMP1,$TMP2,0
      6 106262  001140  001142  001174  ED223:  .WORD   $GDDAT,$BDDAT,$TMP0,0
      7
      8 106272  001134  001140  001136  ED336:  .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
      9
     10 106304  001140  001142  001174  ED337:  .WORD   $GDDAT,$BDDAT,$TMP0,$TMP1,0
     11 106316  001140  001142  001444  ED353:  .WORD   $GDDAT,$BDDAT,RMOF0,0
     12
     13 106326  001336  001346  001350  STSD1:  .WORD   RMCS1I,RMCS2I,RMDSI,RMER1I,RMER2I,RMASI,0
     14 106344  001340  001342  001344  STSD2:  .WORD   RMWCI,RMBAI,RMDAI,RMOFI,RMDCI,RMEC1I
     15 106360  001404  000000          .WORD   RMEC2I,0
     16 106364  001344  001372  001370  STSD3:  .WORD   RMDAI,RMDCI,RMOFI,RMLAI,0
     17 106376  001362  001376  001364  STSD4:  .WORD   RMMR1I,RMMR2I,RMDTI,RMSNI,0
```

```
106410   000                  EF110:  .BYTE   0
106411   000    000           EF111:  .BYTE   0,0
106413   000    000    000    EF114:  .BYTE   0,0,0
106416   000    000    000    EF336:  .BYTE   0,0,0,0
106422   000    000    000    EF337:  .BYTE   0,0,0,0,0

106427   000    000    000    STSF:   .BYTE   0,0,0,0,0,0,0
                              .EVEN
```

```
                                        ;STORAGE FOR GENERAL DATA TRANSFERRS
106436                                  BUFFER:
106436                                  BUFONE: .BLKW   258.
107442                                  BUFTWO: .BLKW   258.

                                        ;STORAGE FOR MANUFACTURES 16 BIT MODE BAD SECTOR FILE
110446  000000  000000                  MFGFIL: .WORD   0,0              ;2 HEADER WORDS
110452                                          .BLKW   256.            ;256. WORDS OF DATA
111452  177777                                  .WORD   -1              ;TERMINATOR IF FILE IS FULL

                                        ;STORAGE FOR USERS 16 BIT MODE BAD SECTOR FILE
111454  000000  000000                  USRFIL: .WORD   0,0              ;2 HEADER WORDS
111460                                          .BLKW   256.            ;256. WORDS OF DATA
112460  177777                                  .WORD   -1              ;TERMINATOR IF FILE IS FULL

        106436                          .=BUFFER

106436                                  HELP:
106436      200                         .ASCII  <CRLF>
106437      200                         .ASCII  <CRLF>
106440      114     111     123         .ASCII  @LIST OF TESTS@<CRLF>
106456      055     055     055         .ASCII  @-------------@<CRLF>
106474      124     061     011         .ASCII  @T1     CONTROLLER ACCESS TEST@<CRLF>
106526      124     062     011         .ASCII  @T2     WRITE, READ ZEROS@<CRLF>
106553      124     063     011         .ASCII  @T3     WRITE, WRITE CHECK ZEROS@<CRLF>
106607      124     064     011         .ASCII  @T4     WRITE, WRITE CHECK ZEROS W/ WCE ERROR@<CRLF>
106660      124     065     011         .ASCII  @T5     WRITE, READ ONES@<CRLF>
106704      124     066     011         .ASCII  @T6     WRITE, WRITE CHECK ONES@<CRLF>
106737      124     067     011         .ASCII  @T7     WRITE, WRITE CHECK ONES W/ WCE ERROR@<CRLF>
107007      124     061     060         .ASCII  @T10    WRITE, WRITE CHECK MULTIPLE SECTORS@<CRLF>
107057      124     061     061         .ASCII  @T11    WRITE, READ W/ IMPLIED SEEK@<CRLF>
107117      124     061     062         .ASCII  @T12    WRITE, WRITE CHECK W/ HEAD SWITCHING@<CRLF>
107170      124     061     063         .ASCII  @T13    WRITE, WRITE CHECK W/ MID-TRANSFER SEEK@<CRLF>
107244      124     061     064         .ASCII  @T14    WRITE, READ W/ HCE ERROR@<CRLF>
107301      124     061     065         .ASCII  @T15    WRITE, READ W/ HCI@<CRLF>
107330      124     061     066         .ASCII  @T16    WRITE, READ W/ IVC ERROR@<CRLF>
107365      124     061     067         .ASCII  @T17    WRITE, READ W/ ABORT@<CRLF>
107416      124     062     060         .ASCII  @T20    WRITE, READ EACH CURRENT LEVEL@<CRLF>
107461      124     062     061         .ASCII  @T21    WRITE, WRITE CHECK W/ CURRENT LEVEL SWITCHING@<CRLF>
107543      124     062     062         .ASCII  @T22    WRITE, READ EARLY PEAK SHIFT@<CRLF>
107604      124     062     063         .ASCII  @T23    WRITE, READ MIXED PEAK SHIFT@<CRLF>
107645      124     062     064         .ASCII  @T24    WRITE, READ EACH TRACK@<CRLF>
107700      124     062     065         .ASCII  @T25    READ, WRITE CHECK MULTIPLE SECTORS IN OFFSET MODE@<CRLF>
107766      200                         .ASCII  <CRLF>
107767      117     120     105         .ASCII  @OPERATIONAL SWITCH SETTINGS@<CRLF>
 10023      055     055     055         .ASCII  @---------------------------@<CRLF>
110057      123     127     111         .ASCII  @SWITCH                   USE@<CRLF>
110074      055     055     055         .ASCII  @------           ----------------------@<CRLF>
110131      040     040     061         .ASCII  @  15             HALT ON ERROR@<CRLF>
110155      040     040     061         .ASCII  @  14             LOOP ON TEST@<CRLF>
110200      040     040     061         .ASCII  @  13             INHIBIT ERROR TYPEOUTS@<CRLF>
110235      040     040     061         .ASCII  @  12             @<CRLF>
110244      0 0     040     061         .ASCII  @  11             INHIBIT ITERATIONS@<CRLF>
110275      040     040     061         .ASCII  @  10             BELL ON ERROR@<CRLF>
110321      040     040     040         .ASCII  @   9             LOOP ON ERROR@<CRLF>
110345      040     040     040         .ASCII  @   8             LOOP ON TEST IN SWR<7:0>@<CRLF>
110404      040     040     040         .ASCII  @   7             TN128@<CRLF>
```

```
     58 110420      040      040      040  .ASCII  @   6          TN64@<CRLF>
     59 110433      040      040      040  .ASCII  @   5          TN32@<CRLF>
     60 110446      040      040      040  .ASCII  @   4          TN16@<CRLF>
     61 110461      040      040      040  .ASCII  @   3          TN8@<CRLF>
     62 110473      040      040      040  .ASCII  @   2          TN4@<CRLF>
     63 110505      040      040      040  .ASCIZ  @   1          TN2@<CRLF>
     64
     65          000200                   .END    200
```

```
ABASE = 176700
ACDW1 = 000000
ACDW2 = 000000
ACKSTS  053432
ACPUOP= 000000
ADDW0 = 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11= 000000
ADDW12= 000000
ADDW13= 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 = 000000
ADDW4 = 000000
ADDW5 = 000000
ADDW6 = 000000
ADDW7 = 000000
ADDW8 = 000000
ADDW9 = 000000
ADEVCT= 000000
ADEVM   000000
ADR   - 000001
AENV  = 000000
AENVM = 000000
AFATAL= 000000
ALL     070514
AMADR1= 000000
AMADR2= 000000
AMADR3= 000000
AMADR4= 000000
AMAMS1= 000000
AMAMS2  000000
AMAMS3  000000
AMAMS4= 000000
AMSGAD= 000000
AMSGLG= 000000
AMSGTY= 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3= 000000
AMTYP4= 000000
AOE   - 001000
APASS = 000000
APE   - 100000
APRIOR= 000000
ATCSU= 000040
APTENV= 000001
APTSIZ= 000200
APTSPD= 000100
ARGS  = 000004
ASNDA   001516
ASNDC   001514
ASWREG= 000000
ATA   = 100000
ATESTN= 000000

ATNMSK  000377
ATNTBL  071746
AUNIT = 000000
AUSWR = 000000
AVECT1= 120254
AVECT2= 000000
A16   = 000400
A17   = 001000
BACK  = 000001
BADSCT  040242
BADTMO  005342
BAI   = 000010
BB00  = 000001
BB01  - 000002
BB02  = 000004
BB03  = 000010
BB04  - 000020
BB05    000040
BB06  = 000100
BB07  - 000200
BB08  - 000400
BB09  = 001000
BIT0  - 000001
BIT00 - 000001
BIT01 - 000002
BIT02 = 000004
BIT03 - 000010
BIT04 = 000020
BIT05 - 000040
BIT06 = 000100
BIT07 - 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 - 040000
BIT15 - 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  - 000040
BIT6  - 000100
BIT7    000200
BIT8  - 000400
BIT9    001000
BLNKS1  071643
BLNKS2  071642
BLNKS3  071641
BLNKS4  071640
BOTADR  066104
BOTFLG  066106
BPTVEC= 000014
BSE     100000
BUFFER  106436

BUFONE  106436
BUFTWO  107442
CC    - 004000
CH    - 002000
CHGADR  001330
CHRCNT  066107
CKSWR = 104410
CLKADR  001520
CLKVCT  001522
CLR   = 000040
CLRSTS  052552
CMNSTA  007476
CMPBUF  042432
CMPERR  050550
CNSL01  071105
CNSL02  071115
CNSL03  071157
CNSL04  071166
CNSL07  071222
CNSL08  071364
CNSL09  071365
CNTCLR  052434
COMMA   070525
CONT  = 000100
CPSAVE  065230
CR    = 000015
CRLF  = 000200
CTLFG   001326
CYLMSK= 001777
DBCK  = 100000
DBEN  = 040000
DBL   = 002000
DCK   = 100000
DDISP = 177570
DEBL  = 020000
DEVSEL  050762
DISPLA  001156
DISPRE  000174
DLT   = 100000
DMD   = 000001
DPE   = 000010
DPEHI = 040000
DPELO = 020000
DPR   = 000400
DRIVES  071600
DRQ   = 004000
DRVCLR= 000010
DRVSTS  055770
DRY   = 000200
DSWR  = 177570
DTASTS  056572
DTE   - 010000
DTO   - 010000
DULPRT= 024024
DVA   - 004000
DVC     000200
EARLY   072272

EBL   - 020000
ECH   = 000100
ECI   = 004000
ECRC  = 001000
EDT1    076614
EDT110  076632
EDT111  076634
EDT114  076636
EDT2    076624
EDT223  076640
EDT336  076650
EDT337  076660
EDT344  076670
EDT353  076702
ED1     106232
ED110   106240
ED111   106244
ED114   106252
ED223   106262
ED336   106272
ED337   106304
ED353   106316
EECC  = 000020
EFT1    076704
EFT110  076722
EFT111  076724
EFT114  076726
EFT2    076714
EFT223  076730
EFT336  076740
EFT337  076750
EFT344  076760
EFT353  076770
EF110   106410
EF111   106411
EF114   106413
EF336   106416
EF337   106422
EHT1    076470
EHT110  076512
EHT111  076516
EHT114  076522
EHT2    076502
EHT223  076526
EHT256  076540
EHT336  076552
EHT337  076564
EHT344  076576
EHT353  076610
EH1     105166
EH110   105205
EH111   105214
EH114   105233
EH223   105262
EH256   105310
EH336   105364
EH337   105423

EH344   105560
EH353   105716
EMS1    076772
EMS10   077326
EMS100  101544
EMS101  101572
EMS102  101620
EMS103  101647
EMS104  101656
EMS105  101705
EMS106  101734
EMS11   077370
EMS110  101753
EMS111  102002
EMS112  102013
EMS113  102077
EMS114  102127
EMS115  102153
EMS116  102156
EMS117  102206
EMS12   077401
EMS120  102230
EMS121  102251
EMS122  102314
EMS123  102351
EMS124  102414
EMS125  102446
EMS126  102511
EMS127  102554
EMS13   077442
EMS130  102613
EMS131  102651
EMS132  102711
EMS133  102735
EMS134  102760
EMS135  103022
EMS136  103065
EMS137  103112
EMS14   077453
EMS140  103147
EMS141  103170
EMS142  103216
EMS143  103233
EMS144  103261
EMS145  103271
EMS146  103320
EMS147  103337
EMS15   077464
EMS150  103354
EMS151  103372
EMS152  103416
EMS153  103442
EMS154  103464
EMS155  103500
EMS156  103520
EMS157  103536
EMS16   077506
```

EMS160	103553	EMS40	100275	EMT121	073716	EMT203	074602	EMT266	075354
EMS161	103570	EMS41	100324	EMT122	073726	EMT204	074614	EMT267	075366
EMS162	103620	EMS42	100352	EMT123	073736	EMT205	074632	EMT27	072630
EMS163	103620	EMS43	100401	EMT124	073746	EMT206	074642	EMT270	075404
EMS164	103644	EMS44	100430	EMT125	073756	EMT207	074652	EMT271	075420
EMS165	103725	EMS45	100503	EMT126	073766	EMT21	072550	EMT272	075436
EMS166	103741	EMS46	100546	EMT127	074000	EMT210	074664	EMT273	075454
EMS167	103766	EMS47	100575	EMT13	072472	EMT211	074672	EMT274	075472
EMS17	077536	EMS5	077164	EMT130	074012	EMT212	074706	EMT275	075510
EMS170	104013	EMS50	100624	EMT131	074024	EMT213	074722	EMT276	075522
EMS171	104042	EMS51	100653	EMT132	074036	EMT214	074732	EMT277	075536
EMS172	104075	EMS52	100701	EMT133	074050	EMT215	074752	EMT3	072416
EMS173	104115	EMS53	100711	EMT134	074062	EMT216	074764	EMT30	072640
EMS174	104132	EMS54	100723	EMT135	074074	EMT217	074774	EMT300	075554
EMS175	104140	EMS55	100745	EMT136	074106	EMT22	072560	EMT301	075574
EMS176	104166	EMS56	100774	EMT137	074120	EMT220	075004	EMT302	075616
EMS177	104215	EMS57	101051	EMT14	072504	EMT221	075014	EMT303	075630
EMS2	077041	EMS6	077214	EMT140	074130	EMT222	075024	EMT304	075642
EMS20	077576	EMS60	101077	EMT141	074140	EMT223	075034	EMT305	075654
EMS200	104237	EMS61	101112	EMT142	074150	EMT224	075044	EMT306	075666
EMS201	104254	EMS62	101141	EMT143	074160	EMT225	075046	EMT307	075676
EMS202	104322	EMS63	101157	EMT144	074170	EMT226	075050	EMT31	072650
EMS203	104352	EMS64	101205	EMT145	074200	EMT227	075052	EMT310	075716
EMS204	104365	EMS65	101223	EMT146	074210	EMT23	072570	EMT311	075730
EMS205	104413	EMS66	101271	EMT147	074220	EMT230	075054	EMT312	075742
EMS206	104425	EMS67	101341	EMT15	072512	EMT231	075056	EMT313	075754
EMS207	104452	EMS7	077261	EMT150	074230	EMT232	075060	EMT314	075774
EMS21	077621	EMS70	101366	EMT151	074240	EMT233	075062	EMT315	076006
EMS210	104477	EMS71	101400	EMT152	074252	EMT234	075064	EMT316	076020
EMS211	104510	EMS72	101413	EMT153	074264	EMT235	075066	EMT317	076032
EMS212	104526	EMS73	101423	EMT154	074302	EMT236	075070	EMT32	072660
EMS213	104551	EMS74	101440	EMT155	074320	EMT237	075072	EMT320	076042
EMS214	104574	EMS75	101457	EMT156	074332	EMT24	072600	EMT321	076052
EMS215	104612	EMS76	101465	EMT157	074344	EMT240	075074	EMT322	076062
EMS216	104633	EMS77	101516	EMT16	072520	EMT241	075076	EMT323	076070
EMS217	104663	EMTVEC=	000030	EMT160	074356	EMT242	075100	EMT324	076100
EMS22	077644	EMT1	072404	EMT161	074366	EMT243	075102	EMT325	076112
EMS220	104716	EMT10	072454	EMT162	074400	EMT244	075104	EMT326	076124
EMS221	104736	EMT100	073452	EMT163	074412	EMT245	075106	EMT327	076142
EMS222	105006	EMT101	073470	EMT164	074422	EMT246	075110	EMT33	072670
EMS223	105037	EMT102	073506	EMT165	074430	EMT247	075120	EMT330	076154
EMS224	105114	EMT103	073516	EMT166	074436	EMT25	072610	EMT331	076164
EMS23	077660	EMT104	073530	EMT167	074444	EMT250	075132	EMT332	076176
EMS24	077706	EMT105	073546	EMT17	072530	EMT251	075146	EMT333	076210
EMS25	077735	EMT106	073556	EMT170	074452	EMT252	075154	EMT334	076226
EMS26	077752	EMT107	073566	EMT171	074464	EMT253	075162	EMT335	076244
EMS27	077773	EMT11	072460	EMT172	074474	EMT254	075200	EMT336	076264
EMS3	077056	EMT110	073606	EMT173	074504	EMT255	075210	EMT337	076274
EMS30	100004	EMT111	073620	EMT174	074514	EMT256	075220	EMT34	072700
EMS31	100014	EMT112	073626	EMT175	074526	EMT257	075230	EMT340	076304
EMS32	100043	EMT113	073634	EMT176	074534	EMT26	072620	EMT341	076316
EMS33	100072	EMT114	073650	EMT177	074542	EMT260	075242	EMT342	076324
EMS34	100120	EMT115	073656	EMT2	072410	EMT261	075254	EMT343	076336
EMS35	100171	EMT116	073666	EMT20	072540	EMT262	075274	EMT344	076350
EMS36	100220	EMT117	073676	EMT200	074544	EMT263	075304	EMT345	076362
EMS37	100247	EMT12	072464	EMT201	074556	EMT264	075314	EMT346	076372
EMS4	077121	EMT120	073706	EMT202	074570	EMT265	075334	EMT347	076406

```
EMT35   072710      ESRC   = 004000      ILRG76- 000076      PACACK= 000022      RMBAI   001342
EMT350  076420      FER    = 000020      IOTVEC= 000020      PAKACK= 000022      RMBAO   001416
EMT351  076436      FIND   = 000001      IPCK0 = 000001      PAR    = 000010     RMCS1  = 000000
EMT352  076446      FMT16  = 010000      IPCK1 = 000002      PAT    = 000020     RMCS1I  001336
EMT353  076452      FNCDTB  071646       IPCK2 = 000004      PDA    = 000400     RMCS10  001412
EMT354  076462      FNCMSK= 000077       IPCK3 = 000010      PFECH   066152       RMCS2  = 000010
EMT36   072720      F0     = 000002      IR     = 000100     PFECH1  066162       RMCS2I  001346
EMT37   072730      F1     = 000004      IVC    = 010000     PFECH2  066250       RMCS20  001422
EMT4    072424      F2     = 000010      LBC    = 002000     PFECH3  066266       RMCS3  = 000052
EMT40   072740      F3     = 000020      LBT    = 002000     PFECH4  066274       RMCS3I  001410
EMT41   072746      F4     = 000040      LF     = 000012     PGE    = 002000     RMCS30  001464
EMT42   072756      GENBUF  042174       LODEV   071506       PGM    = 001000     RMDA   = 000006
EMT43   072770      GET     043060       LS     = 000004     PHA    = 000200     RMDAI   001344
EMT44   073000      GETBUF  001336       LSC    - 004000     PIP    = 020000     RMDAO   001420
EMT45   073010      GETINX  001524       LST    = 000002     PIRQ   = 177772     RMDB   = 000022
EMT46   073020      GETSTS  042774       LSTRK   001334       PIRQVE= 000240      RMDBI   001360
EMT47   073030      GO     = 000001      MCLK   = 004000     PLFS   = 002000     RMDBO   001434
EMT5    072432      GTSWR  = 104407      MCPE   = 020000     PRIERR  044056       RMDC   - 000034
EMT50   073036      HCE    = 000200      MDF    = 000100     PR0    = 000000     RMDCI   001372
EMT51   073046      HCI    = 002000      MDPE   = 000400     PR1    = 000040     RMDCO   001446
EMT52   073060      HCRC    000400       MEDENB  001512       PR2    = 000100     RMDS   - 000012
EMT53   073076      HELP    106436       MFGFIL  110446       PR3    = 000140     RMDSI   001350
EMT54   073114      HT     = 000011      MI     = 000004     PR4    = 000200     RMDSO   001424
EMT55   073124      IAE     002000       MIXED   071756       PR5    = 000240     RMDT   = 000026
EMT56   073136      IBSAVE  065232       MOC    = 000400     PR6    = 000300     RMDTI   001364
EMT57   073154      IDXMSK  000077       MOH    = 020000     PR7    = 000340     RMDTO   001440
EMT6    072440      IE     - 000100      MOL    = 010000     PS     = 177776     RMEC1   000044
EMT60   073164      ILF     000001       MRD    = 002000     PSEL   = 002000     RMEC1I  001402
EMT61   073202      ILF02   000002       MS     - 000040     PSW    = 177776     RMEC10  001456
EMT62   073222      ILF24  = 000024      MSC    = 000002     PUT     043330       RMEC2  = 000046
EMT63   073236      ILF26  - 000026      MSDRVS  071406       PUTBUF  001412       RMEC2I  001404
EMT64   073240      ILF30  = 000030      MSE    = 100000     PUTINX  001553       RMEC20  001460
EMT65   073260      ILF32  - 000032      MSER   = 000200     PWRVEC= 000024      RMER1  - 000014
EMT66   073300      ILF34  - 000034      MSGDRV  071422       QUES    070521       RMER1I  001352
EMT67   073312      ILF36  - 000036      MSHELP  070530       RCLSTS  054226       RMER10  001426
EMT7    072446      ILF40  = 000040      MUR    = 001000     RD     = 000070     RMER2  = 000042
EMT70   073324      ILF42  - 000042      MWD    = 000010     RDCHR  = 104411      RMER2I  001400
EMT71   073334      ILF44  = 000044      MWP    = 000010     RDLIN  = 104412      RMER20  001454
EMT72   073344      ILF46  - 000046      MXF    = 001000     RDOCT  = 104413      RMHR    000036
EMT73   073362      ILF54  - 000054      N       071634       RDY    = 000200     RMHRI   001374
EMT74   073400      ILF56   000056       NDTMSK- 115760       READY   007670       RMHRO   001450
EMT75   073410      ILF64  - 000064      NED    = 010000     RECAL  = 000006     RMLA   - 000020
EMT76   073430      ILF66  = 000066      NEM    = 004000     RESREG= 104415      RMLAI   001356
EMT77   073440      ILF74  = 000074      NONE    071627       RESVEC= 000010      RMLAO   001432
ENRGDT  072404      ILF76  - 000076      NOP    = 000000     REX    = 010000     RMMR1  = 000024
EQUALS  070512      ILR    = 000002      NOTAVL  071540       RG     = 040000     RMMR1I  001362
ERR    - 040000     ILRG50= 000050       NOTPRS  071523       RGDTPT  071756       RMMR10  001436
ERRNMB  066102      ILRG52= 000052       NOTRM   071465       RH     = 000072     RMMR2  = 000040
ERROR  = 104000     ILRG54- 000054       NSA    = 100000     RIP    = 000020     RMMR2I  001376
ERRTYP  065234      ILRG56- 000056       OCC    - 100000     RLEASE= 000012      RMMR20  001452
ERRVEC= 000004      ILRG60= 000060       OFD    = 000200     RMAS   - 000016     RMOF   = 000032
ERTY00  066110      ILRG62= 000062       OFFSET= 000014      RMASI   001354       RMOFI   001370
ERTY01  066115      ILRG64= 000064       OM     - 000001     RMASO   001430       RMOFO   001444
ERTY02  066125      ILRG66= 000066       ONES    072016       RMBA   = 000004     RMR    = 000004
ERTY03  066134      ILRG70= 000070       OPE    = 020000     RMBAE  = 000050     RMSN   = 000030
ERTY04  066142      ILRG72- 000072       OPI     020000       RMBAEI  001406       RMSNI   001360
ERTY05  066145      ILRG74- 000074       OR      000200       RMBAEO  001462       RMSNO   001442
```

SYMBOL TABLE

```
RMWC   = 000002   SW04   = 000020   TST25   035624   $CNTLG  067520   $MADR4  001270
RMWCI    001340   SW05   = 000040   TST3    011274   $CNTLU  067513   $MAIL   001222
RMWCO    001414   SW06   = 000100   TST4    012252   $CPUOP  001250   $MAMS1  001252
RQA    = 100000   SW07   = 000200   TST5    013446   $CRLF   001217   $MAMS2  001256
RQB    = 040000   SW08   = 000400   TST6    014454   $DBLK   063262   $MAMS3  001262
RTC    = 000016   SW09   = 001000   TST7    015432   $DDW0   001306   $MAMS4  001266
R6     =%000006   SW1    = 000002   TYPBN  = 104406   $DDW1   001310   $MBADR  001102
R7     =%000007   SW10   = 002000   TYPDS  = 104405   $DDW2   001312   $MFLG   070422
SADMSK= 000377   SW11   = 004000   TYPE   = 104401   $DDW3   001314   $MNEW   067536
SAVREG= 104414   SW12   = 010000   TYPOC  = 104402   $DDW4   001316   $MSGAD  001236
SA1    = 000001   SW13   = 020000   TYPON  = 104404   $DDW5   001320   $MSGLG  001240
SA16   = 000020   SW14   = 040000   TYPOS  = 104403   $DDW6   001322   $MSGTY  001222
SA2    = 000002   SW15   = 100000   UNS    = 040000   $DDW7   001324   $MSWR   067525
SA4    = 000004   SW2    = 000004   UNTMSK= 000007   $DEVCT  001232   $MTYP1  001253
SA8    = 000010   SW3    = 000010   UNTOFF  071557   $DEVM   001300   $MTYP2  001257
SC     = 100000   SW4    = 000020   UNTON   071570   $DOAGN  037252   $MTYP3  001263
SCOPE  = 000004   SW5    = 000040   UPE    = 020000   $DTBL   063252   $MTYP4  001267
SCTMSG   070426   SW6    = 000100   USE    = 040000   $ENDAD  037242   $MXCNT  064536
SCTMSK= 003700   SW7    = 000200   USRFIL   111454   $ENDCT  037100   $NULL   001170
SC0    = 000100   SW8    = 000400   U0     = 000001   $ENULL  037256   $NWTST= 000001
SC1    = 000200   SW9    = 001000   U1     = 000002   $ENV    001242   $OCNT   063514
SC2    = 000400   SYSTAT   071430   U2     = 000004   $ENVM   001243   $OMODE  063516
SC3    = 001000   TADMSK= 177400   VV     = 000100   $EOP    037044   $OVER   064522
SC4    = 002000   TAG    = 020000   WC     = 000040   $EOPCT  037072   $PASS   001230
SEARCH= 000030   TAGADR   001114   WCD    = 000050   $EOSP   037006   $PASTM  001106
SERERR   044710   TAP      040000   WCE    = 040000   $ERFLG  001117   $POWER  070150
SEEK   - 000004   TA1    - 000400   WCEHI  = 010000   $ERMAX  001131   $PWRDN  070002
SEKSTS   051174   TA16   - 010000   WCELO  = 004000   $ERROR  064640   $PWRMG  070136
SHUT     037262   TA2    - 001000   WCF    = 000040   $ERRPC  001132   $PWRUP  070054
SHUT2    067550   TA4    - 002000   WCH    = 000052   $ERRTB  001602   $QUES   001216
SIZCLK   043550   TA8      004000   WD     = 000060   $ERTTL  001126   $RDCHR  067152
SKI    = 040000   TBITVE= 000014   WH     = 000062   $ESCAP  001210   $RDLIN  067242
SNGPRT- 020024   TIMOUT   043672   WLE    = 004000   $ETABL  001242   $RDOCT  067572
STACK    001100   TKVEC  - 000060   WRL    = 004000   $ETEND  001326   $RDSZ = 000010
STANDA   007020   TPVEC    000064   XSIZ     006464   $FATAL  001224   $RESRE  062734
START    005432   TRAPVE- 000034   XXDP     001332   $FFLG   070424   $RM02   071446
START1   005422   TRE    - 040000   Y        071636   $FILLC  001172   $RM03   071453
START2   005436   TRTVEC- 000014   ZEROS    072060   $FILLS  001171   $RM05   071460
STCDRV   062172   TST    = 010000   $APTHD   001100   $GDADR  001134   $RTNAD  037254
STKLMT= 177774   TSTNMB   066100   $ATYC    070204   $GDDAT  001140   $SAVRE  062676
STOP     064612   TSTPRP   037316   $ATY1    070160   $GET42  037232   $SAVR6  070146
STSD1    106326   TSTQUE   001466   $ATY3    070166   $GTSWR  066700   $SCOPE  064054
STSD2    106344   TST1     010066   $ATY4    070176   $GT42P  037226   $SETUP= 000137
STSD3    106364   TST10    016630   $AUTOB   001150   $HD    = 000000   $STUP = 177777
STSD4    106376   TST11    017634   $BASE    001276   $HIBTS  001100   $SVLAD  064466
STSF     106427   TST12    021060   $BDADR   001136   $HIOCT  067672   $SVPC = 000210
STSH1    105772   TST13    022266   $BDDAT   001142   $ICNT   001120   $SWR  = 167400
STSH2    106047   TST14    023310   $BELL    001212   $ILLUP  070142   $SWREG  001244
STSH3    106136   TST15    025156   $BIN     063044   $INTAG  001151   $SWRMK= 000000
STSH4    106174   TST16    026562   $CDW1    001302   $ITEMB  001130   $SW08T  064540
SWR      001154   TST17    027602   $CDW2    001304   $LF     001220   $TESTN  001226
SWREG    000176   TST2     010266   $CHARC   064050   $LFLG   070423   $TIMES  001206
SW0    = 000001   TST20    030520   $CKSWR   066610   $LPADR  001122   $TKB    001162
SW00   = 000001   TST21    031556   $CMTAG   001114   $LPERR  001124   $TKCNT  066300
SW01   = 000002   TST22    032562   $CM3   - 000000   $MADR1  001254   $TKINT  066310
SW02   = 000004   TST23    033570   $CM4   = 000005   $MADR2  001260   $TKQEN  066307
SW03   - 0000'0   TST24    034576   $CNTLC   067506   $MADR3  001264   $TKQIN  066302
```

```
CZRMOB0 RMOS/3/2 F TNL TST 3    MACRO V04.00  4-APR-81 01:15:59 PAGE 62-6
SYMBOL TABLE                                                          SEQ 0305

$TKQOU  066304          $IMP4   001204          $TRPAD  067746          $TYPEX  064052          $VECT2  001274
$TKQSR  066306          $IN     = 000026        $TSTM   001104          $TYPOC  063316          $XOFF   = 000023
$TKS    001160          $IPB    001166          $TSTNM  001116          $TYPON  063332          $XON    = 000021
$TKSRV  066360          $IPFLG  001173          $TTYIN  067476          $TYPOS  063272          $XTSTR  064076
$TMP0   001174          $IPS    001164          $TYPBN  062772          $UNIT   001234          $$GET4= 000000
$TMP1   001176          $TRAP   067674          $TYPDS  063046          $UNITM  001110          $$SW08= 000026
$TMP2   001200          $TRAP2  067734          $TYPE   063520          $USWR   001246          $OFILL  063515
$TMP3   001202          $TRP    = 000016        $TYPEC  063732          $VECT1  001272          .$X     = 001100


. ABS.  112462      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  62208 WORDS  ( 243 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  70 PAGES
CZRMOB.BIC,CZRMOB/C-CZRMOB.DOC,CZRMOB,SYSMAC/M
```

```
$$GET4  14-20    14-20#
$$SW08  42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1
        42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1
        42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1     42-1
        42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#
        42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#    42-1#
$OFILL  40-1     40-1#    40-1*    40-1*
$40CAT  42-1     43-1
$APTHD   5-8      5-8#
$ASTAT  49-1     49-1
$ATY1   49-1#
$ATY3   41-1     49-1#
$ATY4   43-1     49-1#
$ATYC   49-1     49-1#
$AUTOB   6-0#    10-28*   45-1     45-1     45-1
$BASE    6-0#    10-75    11-33    11-42*   12-57    13-2     13-33    13-180   13-322   13-497   13-645   13-790   13-969   13-:20
        13-=05   13->88   13-@38   13-C24   13-E49   13-F81   13-G99   13-I54   13-K05   13-L50   13-M95   13-046   20-36    21-30
        23-12    28-23    30-15    44-25    60-2
$BDADR   6-0#    13-477*  13-478*  13-479   13-949*  13-950*  13-951   18-84*   60-8
$BDDAT   6-0#    13-465*  13-473*  13-483   13-935*  13-945*  13-955   13-A56*  13-A73*  13-A84*  13-B49*  13-B66*  13-B77*  13-D30*
        13-D31*  13-D34*  13-D37*  13-D92*  13-D93*  13-D96*  13-D99*  13-F10*  13-F72*  13-G27*  13-G82*  18-82*   24-47*   24-48*
        24-51    24-69*   24-75*   24-91*   24-108*  24-124*  24-152*  25-40*   25-41*   25-53*   25-54*   25-64*   25-65*   25-67
        25-85*   25-86*   25-87    25-107*  25-108*  25-109   25-140*  25-141*  25-142   25-156*  25-157*  25-158   25-170*  25-171*
        25-183*  25-184*  25-200*  25-201*  25-226*  25-227*  25-241*  25-242*  25-256*  25-257*  25 271*  25-272*  25-284*  25-285*
        25-314*  25-315*  25-329*  25-330*  25-344*  25-345*  25-361*  25-362*  25-387*  25-388*  25-402*  25-403*  25-417*  25-418*
        25-432*  25-433*  25-455*  25-456*  25-470*  25-471*  25-492*  25-493*  26-10*   26-34*   26-92*   26-93*   26-94    26-120*
        26-121*  26-122   26-137*  26-138   26-150*  26-151   27-39*   27-54*   29-37*   29-66*   29-67*   29-68    29-88*   29-99*
        29-100*  29-106   29-117   29-130*  29-143*  29-168*  29-182*  29-183*  29-196*  29-212*  31-29*   31-30*   31-32    31-42*
        31-51*   31-59    31-69*   31-70*   31-79*   31-80*   31-82    31-92*   31-101*  31-102*  31-104   31-114*  31-115*  31-124*
        31-125*  31-127   32-27*   32-41*   32-57*   32-71*   32-85*   32-99*   32-113*  33-35*   33-49*   33-65*   33-82*   33-103*
        33-120*  33-134*  33-159*  33-174*  33-189*  33-205*  33-222*  33-240*  33-254*  33-270*  34-16*   34-17*   34-19    34-28*
        34-29*   34-31    34-41*   34-50*   34-57*   34-60    34-69*   34-70*   34-72    34-81*   34-82*   34-84    34-94*   34-103*
        35-31*   35-47*   35-67*   35-87*   35-116*  35-135*  35-149*  35-163*  35-189*  35-190*  35-191   35-203*  35-204*  35-240*
        35-256*  35-270*  35-288*  35-301*  35-313*  35-356*  35-369*  35-382*  35-395*  35-417*  35-442*  35-457*  35-470*  35-483*
        35-496*  35-517*  35-534*  35-550*  36-46*   36-62*   36-78*   36-98*   36-112*  60-1     60-6     60-8     60-10    60-11
$BELL   6-0#     43-1     43-1     43-1     45-1     45-1     45-1
$BIN    38-1     38-1#    38-1*    38-1*
$CDW1   6-0#
$CDW2   6-0#
$CHARC  41-1     41-1#    41-1*    41-1*    41-1*
$CKSWR  45-1#    47-1     47-1
$CM3    6-0      6-0#
$CM4    6-0      6-0      6-0      6-0      6-0      6-0      6-0      6-0      6-0      6-0      6-0#     6-0#     6-0#     6-0#
        6-0#     6-0#
$CMTAG  6-0#     10-23    10-23    10-23    10-23    10-23    10-23    10-23
$CNTLC  45-1     45-1     45-1     45-1     45-1#
$CNTLG  45-1     45-1#
$CNTLU  45-1     45-1     45-1#
$CPUOP  6-0#
$CRLF   6-0#     10-48    10-80    10-140   11-20    11-28    11-62    11-100   12-25    12-68    14-20    41-1     41-1     41-1
        43-1     43-1     43-1     44-20    44-61    44-104   44-114   44-126   44-134   44-154   45-1     45-1     45-1     45-1
$DBLK   39-1     39-1     39-1#
$DDW0   6-0#
$DDW1   6-0#
$DDW2   6-0#
$DDW3   6-0#
```

CROSS REFERENCE TABLE (CREF V01-05)

```
$DDW4    6-0#
$DDW5    6-0#
$DDW6    6-0#
$DDW7    6-0#
$DEVCT   6-0#
$DEVM    6-0#    10-67*  10-78   10-117* 10-121* 11-63*  11-70*  11-97*  11-98   12-4    12-38   13-29*  14-28*
$DOAGN  14-20    14-20   14-20#
$DTBL   39-1     39-1#
$ENDAD   5-5     10-28   14-20#  43-1
$ENDCT  10-23    14-20#
$ENULL  14-20#
$ENV     6-0#    10-28   41-1    43-1    49-1    49-1
$ENVM    6-0#    10-23   10-71   41-1    41-1    49-1
$EOP    13-30    14-20#  14-29
$EOPCT  10-23*   14-20   14-20#
$EOSP   14-9#
$ERFLG   6-0#    42-1    42-1    42-1    42-1    42-1    42-1*   43-1    43-1    43-1*
$ERMAX   6-0#    10-23*  42-1    42-1    42-1    42-1*
$ERROR  10-23    43-1#
$ERRPC   6-0#    43-1    43-1    43-1    43-1*   43-1*   44-54
$ERRTB   8-0#    44-72
$ERTTL   6-0#    14-20   14-20   14-20*  43-1    43-1    43-1*
$ESCAP   6-0#    10-23*  42-1*   43-1    43-1    43-1    43-1
$ETABL   6-0#
$ETEND   5-8      6-0#
$FATAL   6-0#    49-1*
$FFLG   49-1     49-1#   49-1*   49-1*   49-1*
$FILLC   6-0#    41-1    41-1    41-1
$FILLS   6-0#    41-1    41-1
$GDADR   6-0#    13-476* 13-479  13-948* 13-951  18-83*  60-8
$GDDAT   6-0#    13-463* 13-464* 13-474* 13-475* 13-483  13-933* 13-934* 13-946* 13-947* 13-955  13-A57* 13-A72* 13-A85*
        13-B50*  13-B65* 13-B78* 13-D26* 13-D88* 13-F11* 13-F12* 13-F70* 13-F71* 13-G25* 13-G26* 13-G80* 13-G81* 18-81*
        24-49*   24-50*  24-51   24-67*  24-68*  24-74*  24-76*  24-88*  24-89*  24-90*  24-106* 24-107* 24-122* 24-123*
        24-150*  24-151* 25-42*  25-55*  25-66*  25-67   25-79*  25-84*  25-87   25-98*  25-106* 25-109  25-135* 25-138*
        25-139*  25-142  25-154* 25-155* 25-158  25-165* 26-16*  26-23*  26-24*  26-25*  26-26*  26-30*  26-32   26-86*
        26-91*   26-94   26-104* 26-117* 26-122  26-134* 26-135* 26-136* 26-138  26-148* 26-149* 26-151  27-40*  27-55*
        29-36*   29-47*  29-63*  29-68   29-87*  29-101* 29-102* 29-129* 29-142* 29-166* 29-167* 29-181* 29-194* 29-195*
        29-210*  29-211* 31-31*  31-32   31-41*  31-57*  31-59   31-68*  31-81*  31-82   31-91*  31-103* 31-104  31-113*
        31-126*  31-127  32-25*  32-26*  32-39*  32-40*  32-58*  32-59*  32-72*  32-73*  32-86*  32-87*  32-100* 32-101*
        32-114*  32-115* 33-33*  33-34*  33-47*  33-48*  33-63*  33-64*  33-80*  33-81*  33-101* 33-102* 33-118* 33-119*
        33-132*  33-133* 33-157* 33-158* 33-173* 33-175* 33-187* 33-188* 33-203* 33-204* 33-220* 33-221* 33-238* 33-239*
        33-252*  33-253* 33-268* 33-269* 34-18*  34-19   34-30*  34-31   34-40*  34-71*  34-72   34-83*  34-84   34-92*
        35-29*   35-30*  35-45*  35-46*  35-65*  35-66*  35-85*  35-86*  35-114* 35-115* 35-133* 35-134* 35-147* 35-148*
        35-161*  35-162* 35-172* 35-187* 35-191  35-205* 35-206* 35-238* 35-239* 35-254* 35-255* 35-268* 35-269* 35-286*
        35-287*  35-299* 35-300* 35-314* 35-315* 35-323* 35-354* 35-355* 35-367* 35-368* 35-380* 35-381* 35-393* 35-394*
        35-415*  35-416* 35-440* 35-441* 35-455* 35-456* 35-468* 35-469* 35-481* 35-482* 35-494* 35-495* 35-515* 35-516*
        35-532*  35-533* 35-548* 35-549* 36-44*  36-45*  36-60*  36-76*  36-77*  36-96*  36-97*  36-110* 36-111* 60-1
        60-6     60-8    60-10   60-11
$GET42  12-34    12-43   14-20#
$GT42P  14-20    14-20#
$GTSWR  45-1#    47-1    47-1
$HD      4-480    4-480    4-480
$HIBTS   5-8#
$HIOCT  46-1#    46-1*
$ICNT    6-0#    42-1    42-1    42-1    42-1*   42-1*
$ILLUP  48-1     48-1    48-1#
$INTAG   6-0#    45-1    45-1    45-1    45-1    45-1*
```

	13-J40#	13-J43#	13-J47#	13-J49#	13-J53#	13-J66	13-J66#	13-J82#	13-J85#	13-J87#	13-J90#	13-J94#	13-J96#	13-K00#
	13-K13	13-K13#	13-K24#	13-K40#	13-K43#	13-K45#	13-K48#	13-K52#	13-K65	13-K65#	13-K82#	13-K85#	13-K87#	13-K90#
	13-K94#	13-K96#	13-K98#	13-L11	13-L11#	13-L27#	13-L30#	13-L32#	13-L35#	13-L39#	13-L41#	13-L43#	13-L45#	13-L58
	13-L58#	13-L69#	13-L85#	13-L88#	13-L90#	13-L93#	13-L97#	13-M10	13-M10#	13-M27#	13-M30#	13-M32#	13-M35#	13-M39#
	13-M41#	13-M43#	13-M56	13-M56#	13-M72#	13-M75#	13-M77#	13-M80#	13-M84#	13-M86#	13-M88#	13-M90#	13-N03	13-N03#
	13-N15#	13-N33#	13-N36#	13-N38#	13-N41#	13-N45#	13-N58	13-N58#	13-N75#	13-N78#	13-N80#	13-N83#	13-N87#	13-N89#
	13-N91#	13-004	13-004#	13-020#	13-023#	13-025#	13-028#	13-032#	13-034#	13-036#	13-038#	13-053	13-053#	13-064#
	13-O20#	13-083#	13-085#	13-088#	13-092#	13-P05	13-P05#	13-P17#	13-P23#	13-P26#	13-P33#	13-P36#	13-P38#	13-P41#
	13-P45#	13-P47#	13-P49#	13-P61	13-P61#	13-P74#	13-P83#	13-P86#	13-P94#	13-P97#	13-P99#	13-Q02#	13-Q06#	13-Q08#
	13-Q10#													
ASNDA	7-0#	16-299*	16-331	16-358*	16-359	16-361*	16-362*	16-363	16-365*	16-397				
ASNDC	7-0#	16-298*	16-330	16-366*	16-367	16-369*	16-396							
ASWREG	6-0	6-0												
ATA	4-552#	25-138	25-139	25-141	29-156	29-157	29-192	29-195	33-145	33-146	33-185	33-188	51-58	51-59
	51-60	51-63	51-64	51-67	51-68	51-69	51-70	51-71	51-72	51-73	51-74	51-75	51-76	51-79
	51-80	51-83	51-84	51-87	51-88									
ATESTN	6-0	6-0												
ATNMSK	4-589#	34-55												
ATNTBL	10-78	10-117	10-121	11-97	12-17	52-3#								
AUNIT	6-0	6-0												
AUSWR	6-0	6-0												
AVECT1	4-773#	6-0	6-0											
AVECT2	6-0	6-0												
BACK	13-41	13-41#	13-52	13-52#	13-68	13-68#	13-71#	13-73#	13-76	13-76#	13-80	13-80#	13-80#	13-93
	13-93#	13-110	13-110#	13-113#	13-115#	13-118	13-118#	13-122	13-122#	13-122#	13-124	13-124#	13-124#	13-126
	13-126#	13-126#	13-139	13-139#	13-155	13-155#	13-158#	13-160#	13-163	13-163#	13-167	13-167#	13-167#	13-169
	13-169#	13-169#	13-171	13-171#	13-171#	13-175	13-175#	13-188	13-188#	13-199	13-199#	13-215	13-215#	13-218#
	13-220#	13-223	13-223#	13-227	13-227#	13-227#	13-240	13-240#	13-257	13-257#	13-260#	13-262#	13-265	13-265#
	13-269	13-269#	13-269#	13-271	13-271#	13-271#	13-273	13-273#	13-273#	13-286	13-286#	13-301	13-301#	13-304#
	13-306#	13-309	13-309#	13-313	13-313#	13-313#	13-315	13-315#	13-315#	13-317	13-317#	13-317#	13-330	13-330#
	13-341	13-341#	13-356	13-356#	13-359#	13-361#	13-364	13-364#	13-368	13-368#	13-368#	13-381	13-381#	13-398
	13-398#	13-401#	13-403#	13-406	13-406#	13-410	13-410#	13-410#	13-412	13-412#	13-412#	13-414	13-414#	13-414#
	13-429	13-429#	13-444	13-444#	13-447#	13-449#	13-452	13-452#	13-456	13-456#	13-456#	13-461	13-461#	13-461#
	13-489	13-489#	13-489#	13-505	13-505#	13-516	13-516#	13-534	13-534#	13-537#	13-539#	13-542	13-542#	13-546
	13-546#	13-546#	13-560	13-560#	13-577	13-577#	13-580#	13-582#	13-585	13-585#	13-589	13-589#	13-589#	13-591
	13-591#	13-591#	13-593	13-593#	13-593#	13-606	13-606#	13-622	13-622#	13-625#	13-627#	13-630	13-630#	13-634
	13-634#	13-634#	13-636	13-636#	13-636#	13-638	13-638#	13-638#	13-640	13-640#	13-653	13-653#	13-664	13-664#
	3-682	13-682#	13-685#	13-687#	13-690	13-690#	13-694	13-694#	13-694#	13-708	13-708#	13-725	13-725#	13-728#
	13-730#	13-733	13-733#	13-737	13-737#	13-737#	13-739	13-739#	13-739#	13-741	13-741#	13-741#	13-754	13-754#
	13-769	13-769#	13-772#	13-774#	13-777	13-777#	13-781	13-781#	13-781#	13-783	13-783#	13-783#	13-785	13-785#
	13-785#	13-798	13-798#	13-809	13-809#	13-827	13-827#	13-830#	13-832#	13-835	13-835#	13-839	13-839#	13-839#
	13-852	13-852#	13-869	13-869#	13-872#	13-874#	13-877	13-877#	13-881	13-881#	13-881#	13-883	13-883#	13-883#
	13-885	13-885#	13-885#	13-900	13-900#	13-915	13-915#	13-918#	13-920#	13-923	13-923#	13-927	13-927#	13-927#
	13-931	13-931#	13-931#	13-961	13-961#	13-961#	13-977	13-977#	13-988	13-988#	13-:06	13-:06#	13-:09#	13-:11#
	13-:14	13-:14#	13-:18	13-:18#	13-:18#	13-:40	13-:40#	13-:55	13-:55#	13-:58#	13-:60#	13-:63	13-:63#	13-:67
	13-:67#	13-:67#	13-:69	13-:69#	13-:69#	13-:71	13-:71#	13-:71#	13-:84	13-:84#	13-:99	13-:99#	13-;02#	13-;04#
	13-;07	13-;07#	13-;11	13-;11#	13-;11#	13-;13	13-;13#	13-;13#	13-;15	13-;15#	13-;15#	13-;28	13-;28#	13-;40
	13-;40#	13-;58	13-;58#	13-;61#	13-;63#	13-;66	13-;66#	13-;70	13-;70#	13-;70#	13-;83	13-;83#	13-;88	13-;88#
	13-;91#	13-;93#	13-;96	13-;96#	13-<00	13-<00#	13-<00#	13-<18	13-<18#	13-<21#	13-<24	13-<24#	13-<28	13-<28#
	13-<28#	13-<30	13-<30#	13-<30#	13-<32	13-<32#	13-<32#	13-<45	13-<45#	13-<50	13-<50#	13-<53#	13-<55#	13-<58
	13-<58#	13-<62	13-<62#	13-<62#	13-<79	13-<79#	13-<82#	13-<85	13-<85#	13-<89	13-<89#	13-<89#	13-<91	13-<91#
	13-<91#	13-<93	13-<93#	13-<93#	13-<95	13-<95#	13-=20	13- 20#	13-=29	13-=29#	13-=43	13-=43#	13-=46#	13-=48#
	13-=51	13-=51#	13-=55	13-=55#	13-=55#	13-~76	13-=76#	13-~80	13-~80#	13-~83#	13-~85#	13-=88	13-=88#	13-~92
	13-=92#	13-=92#	13->07	13->07#	13->10#	13->12#	13->15	13->15#	13->19	13->19#	13->19#	13->21	13->21#	13->21#
	13->23	13->23#	13->23#	13->36	13->36#	13->40	13->40#	13->43#	13->45#	13->48	13->48#	13->52	13->52#	13->52#
	13->67	13->67#	13->70#	13->72#	13->75	13->75#	13->79	13->79#	13->79#	13->81	13->81#	13->81#	13->83	13->83#
	13->83#	13-?03	13-?03#	13-?12	13-?12#	13-?25	13-?25#	13-?28#	13-?30#	13-?33	13-?33#	13-?37	13-?37#	13-?37#

EH111 55-5 59-3#
EH114 55-7 59-5#
EH223 55-8 59-6#
EH256 55-9 59-7#
EH336 55-11 59-10#
EH337 55-12 59-11#
EH344 55-13 59-15#
EH353 55-15 59-18#
EHT1 8-3 8-6 8-9 8-12 8-15 8-18 8-24 8-27 8-30 8-33 8-36 8-39 8-42 8-45
8-48 8-51 8-54 8-57 8-60 8-63 8-66 8-69 8-72 8-75 8-78 8-81 8-84 8-87
8-90 8-93 8-96 8-99 8-102 8-105 8-108 8-111 8-114 8-117 8-120 8-123 8-126 8-130
8-133 8-136 8-139 8-142 8-146 8-150 8-154 8-160 8-163 8-166 8-169 8-172 8-175 8-178
8-182 8-185 8-188 8-191 8-194 8-197 8-200 8-203 8-206 8-209 8-212 8-215 8-218 8-236
8-239 8-242 8-245 8-248 8-251 8-254 8-257 8-260 8-263 8-266 8-269 8-272 8-275 8-278
8-281 8-284 8-287 8-290 8-293 8-296 8-299 8-302 8-305 8-308 8-311 8-314 8-317 8-320
8-323 8-326 8-329 8-332 8-335 8-338 8-341 8-344 8-347 8-354 8-357 8-360 8-363 8-366
8-369 8-372 8-375 8-378 8-381 8-384 8-390 8-393 8-396 8-399 8-402 8-405 8-408 8-411
8-414 8-417 8-420 8-423 8-432 8-435 8-438 8-441 8-444 8-504 8-507 8-510 8-519 8-522
8-525 8-531 8-534 8-537 8-540 8-543 8-546 8-549 8-552 8-556 8-559 8-563 8-566 8-569
8-573 8-577 8-581 8-586 8-590 8-594 8-597 8-600 8-603 8-606 8-609 8-612 8-615 8-618
8-621 8-624 8-627 8-630 8-633 8-636 8-639 8-642 8-645 8-648 8-651 8-654 8-657 8-660
8-663 8-666 8-669 8-672 8-675 8-678 8-693 8-696 8-702 8-705 8-708 8-711 8-714 8-723
55-1#
EHT110 8-221 55-4#
EHT111 8-224 8-227 55-5#
EHT114 8-233 55-7#
EHT2 8-426 8-429 8-513 8-516 55-2#
EHT223 8-447 55-8#
EHT256 8-528 55-9#
EHT336 8-351 8-681 8-687 8-690 55-11#
EHT337 8-684 55-12#
EHT344 8-699 55-13#
EHT353 8-720 55-15#
EMS1 54-3 58-3#
EMS10 54-7 54-11 58-10#
EMS100 54-39 54-71 54-176 54-196 58-66#
EMS101 54-40 54-59 54-174 54-197 58-67#
EMS102 54-177 54-178 54-181 54-182 54-183 54-184 54-186 54-187 54-188 54-189 54-234 54-235 58-68#
EMS103 54-56 54-57 54-135 54-141 54-196 54-197 54-198 54-203 54-204 54-205 54-206 54-207 54-208 54-209
54-210 54-211 54-213 54-214 54-215 54-219 54-220 58-69#
EMS104 54-136 54-180 54-203 58-70#
EMS105 54-204 58-71#
EMS106 54-212 54-232 58-72#
EMS11 54-12 54-15 54-16 54-17 54-18 54-19 54-20 54-21 54-22 54-23 54-24 54-25 54-26 54-27
54-28 54-29 54-30 54-31 54-32 54-33 54-43 54-177 58-11#
EMS110 54-9 54-74 54-77 58-73#
EMS111 54-74 54-226 58-74#
EMS112 54-74 58-75#
EMS113 54-75 54-76 54-78 54-127 58-77#
EMS114 54-77 58-78#
EMS115 54-44 54-45 54-48 54-50 54-51 54-52 54-54 54-55 54-60 54-61 54-63 54-66 54-67 54-70
54-73 54-77 54-107 54-108 54-109 54-110 54-134 54-137 54-139 54-140 54-142 54-179 54-182 54-183
54-185 54-186 54-187 54-188 54-189 54-190 54-192 54-193 54-194 54-195 54-201 54-205 54-221 54-222
54-223 54-232 54-234 58-79#
EMS116 54-74 54-77 54-191 54-225 58-80#
EMS117 54-77 54-143 54-148 54-149 54-178 54-185 54-201 54-225 54-228 54-229 54-230 54-231 58-81#
EMS12 54-12 58-12#

```
EMS120  54-78   58-82#
EMS121  54-79   54-88   54-97   54-99   58-83#
EMS122  54-80   54-89   58-84#
EMS123  54-81   54-88   54-89   54-90   54-90   54-91   54-92   54-93   54-94   54-95   54-96   54-98   54-126  58-85#
EMS124  54-56   54-82   54-91   54-101  58-86#
EMS125  54-83   54-92   54-102  54-169  58-87#
EMS126  54-84   54-93   54-103  58-88#
EMS127  58-89#
EMS13   54-13   54-42   54-216  58-13#
EMS130  54-86   54-95   54-106  58-90#
EMS131  54-87   54-96   54-104  58-91#
EMS132  54-79   54-80   54-81   54-82   54-83   54-84   54-85   54-86   54-87   54-88   54-89   54-90   54-91   54-92
        54-93   54-94   54-95   54-96   54-97   54-98   54-99   54-100  54-101  54-102  54-103  54-104  54-105  54-106
        54-125  54-126  54-168  54-169  58-92#
EMS133  54-79   54-80   54-81   54-82   54-83   54-84   54-85   54-86   54-87   54-125  58-93#
EMS134  54-88   54-89   54-90   54-91   54-92   54-93   54-94   54-95   54-96   54-126  58-94#
EMS135  54-97   54-98   58-95#
EMS136  54-99   54-100  54-101  54-102  54-103  54-104  54-105  54-106  58-96#
EMS137  54-85   54-94   54-100  58-97#
EMS14   54-13   58-14#
EMS140  54-44   54-52   54-60   54-107  54-117  54-137  54-186  54-192  58-98#
EMS141  54-35   54-44   54-45   54-60   54-61   54-66   54-107  54-122  54-137  54-187  54-188  54-193  54-222  58-99#
EMS142  54-47   54-65   54-108  54-140  54-184  54-202  58-100#
EMS143  54-47   54-65   54-73   54-108  54-109  54-110  54-133  54-140  54-184  54-202  54-205  58-101#
EMS144  54-36   54-73   54-109  54-110  54-205  58-102#
EMS145  54-36   54-73   54-109  54-110  54-205  58-103#
EMS146  54-109  54-185  54-190  54-201  58-104#
EMS147  54-69   54-112  54-113  54-173  54-217  58-105#
EMS15   54-13   54-42   54-216  58-15#
EMS150  54-50   54-51   54-54   54-63   54-67   54-111  54-134  54-138  54-182  54-183  54-194  54-195  54-223  54-234
        58-106#
EMS151  54-69   54-112  54-113  54-173  54-217  58-107#
EMS152  54-50   54-51   54-54   54-63   54-111  54-134  54-182  54-183  54-194  54-195  54-223  58-108#
EMS153  54-111  58-109#
EMS154  54-111  54-112  54-123  54-124  58-110#
EMS155  54-112  54-124  58-111#
EMS156  54-113  54-114  54-115  54-116  54-173  58-112#
EMS157  54-113  54-123  58-113#
EMS16   54-13   58-16#
EMS160  54-114  54-115  54-116  58-114#
EMS161  54-114  58-115#
EMS162  58-116#
EMS163  54-58   54-59   54-60   54-61   54-62   54-63   54-64   54-65   54-66   54-67   54-68   54-69   54-70   54-71
        54-72   54-73   54-117  54-143  58-117#
EMS164  54-45   58-118#
EMS165  54-56   54-57   54-71   54-72   54-172  54-196  54-197  54-198  58-120#
EMS166  54-142  58-121#
EMS167  54-8    54-49   54-68   54-123  54-124  54-168  54-169  54-170  54-175  54-235  58-122#
EMS17   54-14   58-17#
EMS170  54-9    58-123#
EMS171  54-57   54-105  54-125  54-126  58-124#
EMS172  54-215  58-125#
EMS173  54-214  58-126#
EMS174  54-216  58-127#
EMS175  54-216  54-217  54-218  54-219  58-128#
EMS176  58-129#
EMS177  54-127  58-130#
```

```
EMS2    54-4    54-5    58-4#
EMS20   54-34   54-58   54-110  54-118  54-119  54-120  54-121  54-212  54-232  54-238  58-18#
EMS200  54-128  58-131#
EMS201  54-128  58-132#
EMS202  54-122  54-130  54-131  54-132  54-133  54-134  54-136  54-180  58-133#
EMS203  54-122  54-130  54-131  54-132  54-133  54-174  54-175  54-176  54-237  58-134#
EMS204  54-131  58-135#
EMS205  54-55   54-142  54-179  54-233  58-136#
EMS206  54-55   54-142  54-170  54-179  54-233  58-137#
EMS207  54-168  54-170  58-138#
EMS21   54-15   58-19#
EMS210  54-169  54-170  54-173  58-139#
EMS211  54-170  58-140#
EMS212  54-171  54-172  58-141#
EMS213  54-171  58-142#
EMS214  54-55   58-143#
EMS215  54-236  58-144#
EMS216  54-237  58-145#
EMS217  54-237  58-146#
EMS22   54-16   58-20#
EMS220  54-179  58-147#
EMS221  54-231  54-238  58-148#
EMS222  54-233  58-149#
EMS223  54-232  58-150#
EMS224  54-121  58-151#
EMS23   54-17   58-21#
EMS24   54-18   58-22#
EMS25   54-19   54-115  58-23#
EMS26   54-20   54-116  54-173  58-24#
EMS27   54-16   54-19   54-20   54-115  54-116  54-173  58-25#
EMS3    54-4    54-76   58-5#
EMS30   54-17   54-18   54-21   54-22   54-23   54-24   54-25   54-26   54-27   54-28   54-29   54-30   54-31   54-3c
        54-33   54-43   54-122  54-130  54-131  54-132  54-133  54-144  54-145  54-146  54-147  54-174  54-175  54-176
        54-177  54-199  58-26#
EMS31   54-21   54-146  54-207  54-208  58-27#
EMS32   54-22   58-28#
EMS33   54-23   54-219  58-29#
EMS34   54-24   54-225  54-226  54-227  58-30#
EMS35   54-25   54-216  58-31#
EMS36   54-26   54-213  54-214  54-215  58-32#
EMS37   54-27   58-33#
EMS4    54-5    54-75   58-6#
EMS40   54-28   54-220  58-34#
EMS41   54-29   54-190  54-191  58-35#
EMS42   54-30   54-206  58-36#
EMS43   54-31   54-209  58-37#
EMS44   54-32   54-211  54-230  58-38#
EMS45   54-33   54-210  54-229  58-39#
EMS46   54-43   54-58   54-120  54-145  54-177  54-185  54-199  54-201  58-40#
EMS47   54-37   54-44   54-45   54-52   54-60   54-61   54-66   54-117  54-118  54-147  54-186  54-187  54-188  54-192
        54-193  54-222  58-41#
EMS5    54-6    54-7    58-7#
EMS50   54-46   54-48   54-64   54-70   54-132  54-135  54-139  54-141  54-181  54-185  54-189  54-2    54-201  54-221
        58-42#
EMS51   54-48   54-55   54-70   54-130  54-139  54-142  54-179  54-189  54-221  58-43#
EMS52   54-55   54-142  54-143  54-148  54-149  54-178  54-179  54-185  54-201  54-228  54-229  54-230  54-231  54-233
        58-44#
```

EMS53	54-37	54-38	54-39	54-40	54-44	54-45	54-46	54-47	54-62	54-64	54-65	54-73	54-140	54-141
	54-181	54-182	54-184	54-186	54-187	54-190	54-191	54-192	54-193	54-194	54-195	54-200	54-202	54-203
	54-204	54-206	54-207	54-216	54-218	54-219	54-220	54-226	58-45#					
EMS54	54-226	58-46#												
EMS55	54-227	58-47#												
EMS56	54-50	54-51	54-54	54-62	54-63	54-67	54-119	54-143	54-144	54-178	54-182	54-183	54-194	54-195
	54-223	54-228	54-234	58-48#										
EMS57	54-190	54-191	54-192	54-193	54-194	54-195	54-196	54-197	54-198	54-199	54-200	54-201	54-202	54-203
	54-204	54-205	54-206	54-207	54-208	54-209	54-210	54-211	54-213	54-214	54-215	54-220	54-221	54-222
	54-223	54-228	54-229	54-230	58-49#									
EMS6	54-6	58-8#												
EMS60	54-224	58-50#												
EMS61	54-224	58-51#												
EMS62	54-224	58-52#												
EMS63	54-42	54-149	54-190	58-53#										
EMS64	54-8	54-14	54-41	54-144	54-145	54-146	54-147	54-199	58-54#					
EMS65	54-41	58-55#												
EMS66	54-34	54-218	58-56#											
EMS67	54-43	54-44	54-45	54-46	54-47	54-48	54-49	54-50	54-51	54-52	54-54	58-57#		
EMS7	54-10	58-9#												
EMS70	54-48	54-50	54-52	54-54	54-66	54-70	54-73	54-107	54-110	54-139	54-182	54-189	54-194	54-205
	54-221	58-58#												
EMS71	54-51	54-59	54-63	54-183	54-195	54-208	58-59#							
EMS72	54-36	54-48	54-51	54-55	54-56	54-57	54-60	54-61	54-61	54-63	54-108	54-109	54-134	54-135
	54-139	54-140	54-142	54-179	54-183	54-187	54-193	54-195	54-232	58-60#				
EMS73	54-49	54-54	54-67	54-68	54-136	54-180	54-188	54-222	54-223	54-234	54-235	58-61#		
EMS74	54-69	54-70	54-134	54-189	54-217	54-221	58-62#							
EMS75	54-35	54-66	54-67	54-137	54-138	54-188	54-222	54-223	54-234	58-63#				
EMS76	54-35	54-36	54-37	54-38	54-39	54-40	58-64#							
EMS77	54-38	54-72	54-148	54-198	58-65#									
EMT1	8-3	54-3#												
EMT10	8-24	54-10#												
EMT100	8-197	54-66#												
EMT101	8-200	54-67#												
EMT102	8-203	54-68#												
EMT103	8-206	54-69#												
EMT104	8-209	54-70#												
EMT105	8-212	54-71#												
EMT106	8-215	54-72#												
EMT107	8-218	54-73#												
EMT11	8-27	54-11#												
EMT110	8-221	54-74#												
EMT111	8-224	54-75#												
EMT112	8-227	54-76#												
EMT113	8-230	54-77#												
EMT114	8-233	54-78#												
EMT115	8-236	54-79#												
EMT116	8-239	54-80#												
EMT117	8-242	54-81#												
EMT12	8-30	54-12#												
EMT120	8-245	54-82#												
EMT121	8-248	54-83#												
EMT122	8-251	54-84#												
EMT123	8-254	54-85#												
EMT124	8-257	54-86#												
EMT125	8-260	54-87#												
EMT126	8-263	54-88#												

EMT127 8-266 54-89#
EMT13 8-33 54-13#
EMT130 8-269 54-90#
EMT131 8-272 54-91#
EMT132 8-275 54-92#
EMT133 8-278 54-93#
EMT134 8-281 54-94#
EMT135 8-284 54-95#
EMT136 8-287 54-96#
EMT137 8-290 54-97#
EMT14 8-36 54-14#
EMT140 8-293 54-98#
EMT141 8-296 54-99#
EMT142 8-299 54-100#
EMT143 8-302 54-101#
EMT144 8-305 54-102#
EMT145 8-308 54-103#
EMT146 8-311 54-104#
EMT147 8-314 54-105#
EMT15 8-39 54-15#
EMT150 8-317 54-106#
EMT151 8-320 54-107#
EMT152 8-323 54-108#
EMT153 8-326 54-109#
EMT154 8-329 54-110#
EMT155 8-332 54-111#
EMT156 8-335 54-112#
EMT157 8-338 54-113#
EMT16 8-42 54-16#
EMT160 8-341 54-114#
EMT161 8-344 54-115#
EMT162 8-347 54-116#
EMT163 54-117#
EMT164 8-354 54-118#
EMT165 8-357 54-119#
EMT166 8-360 54-120#
EMT167 8-363 54-121#
EMT17 8-45 54-17#
EMT170 8-366 54-122#
EMT171 8-369 54-123#
EMT172 8-372 54-124#
EMT173 8-375 54-125#
EMT174 8-378 54-126#
EMT175 8-381 54-127#
EMT176 8-384 54-128#
EMT177 54-129#
EMT2 8-6 54-4#
EMT20 8-48 54-18#
EMT200 8-390 54-130#
EMT201 8-393 54-131#
EMT202 8-396 54-132#
EMT203 8-399 54-133#
EMT204 8-402 54-134#
EMT205 8-405 54-135#
EMT206 8-408 54-136#
EMT207 8-411 54-137#
EMT21 8-51 54-19#

EMT210 8-414 54-138#
EMT211 8-417 54-139#
EMT212 8-420 54-140#
EMT213 8-423 54-141#
EMT214 8-426 54-142#
EMT215 8-429 54-143#
EMT216 8-432 54-144#
EMT217 8-435 54-145#
EMT22 8-54 54-20#
EMT220 8-438 54-146#
EMT221 8-441 54-147#
EMT222 8-444 54-148#
EMT223 8-447 54-149#
EMT224 54-150#
EMT225 54-151#
EMT226 54-152#
EMT227 54-153#
EMT23 8-57 54-21#
EMT230 54-154#
EMT231 54-155#
EMT232 54-156#
EMT233 54-157#
EMT234 54-158#
EMT235 54-159#
EMT236 54-160#
EMT237 54-161#
EMT24 8-60 54-22#
EMT240 54-162#
EMT241 54-163#
EMT242 54-164#
EMT243 54-165#
EMT244 54-166#
EMT245 54-167#
EMT246 8-504 54-168#
EMT247 8-507 54-169#
EMT25 8-63 54-23#
EMT250 8-510 54-170#
EMT251 8-513 54-171#
EMT252 8-516 54-172#
EMT253 8-519 54-173#
EMT254 8-522 54-174#
EMT255 8-525 54-175#
EMT256 8-528 54-176#
EMT257 8-531 54-177#
EMT26 8-66 54-24#
EMT260 8-534 54-178#
EMT261 8-537 54-179#
EMT262 8-540 54-180#
EMT263 8-543 54-181#
EMT264 8-546 54-182#
EMT265 8-549 54-183#
EMT266 8-552 54-184#
EMT267 8-555 54-185#
EMT27 8-69 54-25#
EMT270 8-559 54-186#
EMT271 8-563 54-187#
EMT272 8-566 54-188#

EMT273	8-569	54-189#	
EMT274	8-573	54-190#	
EMT275	8-577	54-191#	
EMT276	8-581	54-192#	
EMT277	8-586	54-193#	
EMT3	8-9	54-5#	
EMT30	8-72	54-26#	
EMT300	8-590	54-194#	
EMT301	8-594	54-195#	
EMT302	8-597	54-196#	
EMT303	8-600	54-197#	
EMT304	8-603	54-198#	
EMT305	8-606	54-199#	
EMT306	8-609	54-200#	
EMT307	8-612	54-201#	
EMT31	8-75	54-27#	
EMT310	8-615	54-202#	
EMT311	8-618	54-203#	
EMT312	8-621	54-204#	
EMT313	8-624	54-205#	
EMT314	8-627	54-206#	
EMT315	8-630	54-207#	
EMT316	8-633	54-208#	
EMT317	8-636	54-209#	
EMT32	8-78	54-28#	
EMT320	8-639	54-210#	
EMT321	8-642	54-211#	
EMT322	8-645	54-212#	
EMT323	8-648	54-213#	
EMT324	8-651	54-214#	
EMT325	8-654	54-215#	
EMT326	8-657	54-216#	
EMT327	8-660	54-217#	
EMT33	8-81	54-29#	
EMT330	8-663	54-218#	
EMT331	8-666	54-219#	
EMT332	8-669	54-220#	
EMT333	8-672	54-221#	
EMT334	8-675	54-222#	
EMT335	8-678	54-223#	
EMT336	8-351	8-681	54-224#
EMT337	8-684	54-225#	
EMT34	8-84	54-30#	
EMT340	8-687	54-226#	
EMT341	8-690	54-227#	
EMT342	8-693	54-228#	
EMT343	8-696	54-229#	
EMT344	8-699	54-230#	
EMT345	8-702	54-231#	
EMT346	8-705	54-232#	
EMT347	8-708	54-233#	
EMT35	8-87	54-31#	
EMT350	8-711	54-234#	
EMT351	8-714	54-235#	
EMT352	8-717	54-236#	
EMT353	8-720	54-237#	
EMT354	8-723	54-238#	

```
F0      4-495#  25-29
F1      4-494#  25-29
F2      4-493#  25-29   26-113
F3      4-492#  25-29
F4      4-491#  25-29
FER     4-577#   4-583  13-A67  13-A72  13-B60  13-B65  13-D28  13-D31  13-D90  13-D93  16-222  25-416  35-348  35-365
       35-368  35-391  35-394
FIND   13-41   13-41#  13-41#  13-52   13-52#  13-52#  13-68   13-68#  13-68#  13-71   13-71#  13-73   13-73#  13-76
       13-76#  13-76#  13-80   13-80#  13-80#  13-93   13-93#  13-93#  13-110  13-110#  13-110#  13-113  13-113#  13-115
       13-115#  13-118  13-118#  13-118#  13-122  13-122#  13-122#  13-124  13-124#  13-124#  13-126  13-126#  13-126#  13-139
       13-139#  13-139#  13-155  13-155#  13-155#  13-158  13-158#  13-160  13-160#  13-163  13-163#  13-163#  13-167  13-167#
       13-167#  13-169  13-169#  13-169#  13-171  13-171#  13-171#  13-175  13-175#  13-175#  13-188  13-188#  13-188#  13-199
       13-199#  13-199#  13-215  13-215#  13-215#  13-218  13-218#  13-220  13-220#  13-223  13-223#  13-223#  13-227  13-227#
       13-227#  13-240  13-240#  13-240#  13-257  13-257#  13-257#  13-260  13-260#  13-262  13-262#  13-265  13-265#  13-265#
       13-269  13-269#  13-269#  13-271  13-271#  13-271#  13-273  13-273#  13-273#  13-286  13-286#  13-286#  13-301  13-301#
       13-301#  13-304  13-304#  13-306  13-306#  13-309  13-309#  13-309#  13-313  13-313#  13-313#  13-315  13-315#  13-315#
       13-317  13-317#  13-317#  13-330  13-330#  13-330#  13-341  13-341#  13-341#  13-356  13-356#  13-356#  13-359  13-359#
       13-361  13-361#  13-364  13-364#  13-364#  13-368  13-368#  13-368#  13-381  13-381#  13-381#  13-398  13-398#  13-398#
       13-401  13-401#  13-403  13-403#  13-406  13-406#  13-406#  13-410  13-410#  13-410#  13-412  13-412#  13-412#  13-414
       13-414#  13-414#  13-429  13-429#  13-429#  13-444  13-444#  13-444#  13-447  13-447#  13-449  13-449#  13-452  13-452#
       13-452#  13-456  13-456#  13-456#  13-461  13-461#  13-461#  13-489  13-489#  13-489#  13-505  13-505#  13-505#  13-516
       13-516#  13-516#  13-534  13-534#  13-534#  13-537  13-537#  13-539  13-539#  13-542  13-542#  13-542#  13-546  13-546#
       13-546#  13-560  13-560#  13-560#  13-577  13-577#  13-577#  13-580  13-580#  13-582  13-582#  13-585  13-585#  13-585#
       13-589  13-589#  13-589#  13-591  13-591#  13-591#  13-593  13-593#  13-593#  13-606  13-606#  13-606#  13-622  13-622#
       13-622#  13-625  13-625#  13-627  13-627#  13-630  13-630#  13-630#  13-634  13-634#  13-634#  13-636  13-636#  13-636#
       13-638  13-638#  13-638#  13-640  13-640#  13-640#  13-653  13-653#  13-653#  13-664  13-664#  13-664#  13-682  13-682#
       13-682#  13-685  13-685#  13-687  13-687#  13-690  13-690#  13-690#  13-694  13-694#  13-694#  13-708  13-708#  13-708#
       13-725  13-725#  13-725#  13-728  13-728#  13-730  13-730#  13-733  13-733#  13-733#  13-737  13-737#  13-737#  13-739
       13-739#  13-739#  13-741  13-741#  13-741#  13-754  13-754#  13-754#  13-769  13-769#  13-769#  13-772  13-772#  13-774
       13-774#  13-777  13-777#  13-777#  13-781  13-781#  13-781#  13-783  13-783#  13-783#  13-785  13-785#  13-785#  13-798
       13-798#  13-798#  13-809  13-809#  13-809#  13-827  13-827#  13-827#  13-830  13-830#  13-832  13-832#  13-835  13-835#
       13-835#  13-839  13-839#  13-839#  13-852  13-852#  13-852#  13-869  13-869#  13-869#  13-872  13-872#  13-874  13-874#
       13-877  13-877#  13-877#  13-881  13-881#  13-881#  13-883  13-883#  13-883#  13-885  13-885#  13-885#  13-900  13-900#
       13-900#  13-915  13-915#  13-915#  13-918  13-918#  13-920  13-920#  13-923  13-923#  13-923#  13-927  13-927#  13-927#
       13-931  13-931#  13-931#  13-961  13-961#  13-961#  13-977  13-977#  13-977#  13-988  13-988#  13-988#  13-:06  13-:06#
       13-:06#  13-:09  13-:09#  13-:11  13-:11#  13-:14  13-:14#  13-:14#  13-:18  13-:18#  13-:18#  13-:40  13-:40#  13-:40#
       13-:55  13-:55#  13-:55#  13-:58  13-:58#  13-:60  13-:60#  13-:63  13-:63#  13-:63#  13-:67  13-:67#  13-:67#  13-:69
       13-:69#  13-:69#  13-:71  13-:71#  13-:71#  13-:84  13-:84#  13-:84#  13-:99  13-:99#  13-:99#  13-;02  13-;02#  13-;04
       13-;04#  13-;07  13-;07#  13-;07#  13-;11  13-;11#  13-;11#  13-;13  13-;13#  13-;13#  13-;15  13-;15#  13-;15#  13-;28
       13-;28#  13-;28#  13-;40  13-;40#  13-;40#  13-;58  13-;58#  13-;58#  13-;61  13-;61#  13-;63  13-;63#  13-;66  13-;66#
       13-;66#  13-;70  13-;70#  13-;70#  13-;83  13-;83#  13-;83#  13-;88  13-;88#  13-;88#  13-;91  13-;91#  13-;93  13-;93#
       13-;96  13-;96#  13-;96#  13-<00  13-<00#  13-<00#  13-<18  13-<18#  13-<18#  13-<21  13-<21#  13-<24  13-<24#  13-<24#
       13-<28  13-<28#  13-<28#  13-<30  13-<30#  13-<30#  13-<32  13-<32#  13-<32#  13-<45  13-<45#  13-<45#  13-<50  13-<50#
       13-<50#  13-<53  13-<53#  13-<55  13-<55#  13-<58  13-<58#  13-<58#  13-<62  13-<62#  13-<62#  13-<79  13-<79#  13-<79#
       13-<82  13-<82#  13-<85  13-<85#  13-<85#  13-<89  13-<89#  13-<89#  13-<91  13-<91#  13-<91#  13-<93  13-<93#  13-<93#
       13-<95  13-<95#  13-<95#  13-=20  13-=20#  13-=20#  13-_29  13-=29#  13-=29#  13-=43  13-=43#  13-=43#  13-=46  13-=46#
       13-=48  13-=48#  13-_51  13-_51#  13-_51#  13-=55  13-=55#  13-=55#  13-=76  13-=76#  13-=76#  13-=80  13-=80#  13-=80#
       13-=83  13-=83#  13-=85  13-=85#  13-=88  13-=88#  13-_88#  13-_92  13-=92#  13-=92#  13->07  13->07#  13->07#  13->10
       13->10#  13->12  13->12#  13->15  13->15#  13->15#  13->19  13->19#  13->19#  13->21  13->21#  13->21#  13->23  13->23#
       13->23#  13->36  13->36#  13->36#  13->40  13->40#  13->40#  13->43  13->43#  13->45  13->45#  13->48  13->48#  13->48#
       13->52  13->52#  13->52#  13->67  13->67#  13->67#  13->70  13->70#  13->72  13->72#  13->75  13->75#  13->75#  13->79
       13->79#  13->79#  13->81  13->81#  13->81#  13->83  13->83#  13->83#  13-?03  13-?03#  13-?03#  13-?12  13-?12#  13-?12#
       13-?25  13-?25#  13-?25#  13-?28  13-?28#  13-?30  13-?30#  13-?33  13-?33#  13-?33#  13-?37  13-?37#  13-?37#  13-?58
       13-?58#  13-?58#  13-?73  13-?73#  13-?73#  13-?76  13-?76#  13-?78  13-?78#  13-?81  13-?81#  13-?81#  13-?85  13-?85#
       13-?85#  13-?87  13-?87#  13-?87#  13-?89  13-?89#  13-?89#  13-@02  13-@02#  13-@02#  13-@17  13-@17#  13-@17#  13-@20
       13-@20#  13-@22  13-@22#  13-@25  13-@25#  13-@25#  13-@29  13-@29#  13-@29#  13-@31  13-@31#  13-@31#  13-@33  13-@33#
       13-@33#  13-@48  13-@48#  13-@48#  13-@59  13-@59#  13-@59#  13-@83  13-@83#  13-@83#  13-@86  13-@86#  13-@88  13-@88#
```

	13-L35	13-L93	13-M35	13-M80	13-N41	13-N83	13-028	13-088	13-P41	13-Q02	15-47	15-90	15-114	15-144
	15-168	15-199	16-76	16-97	16-129	16-163	16-192	20-31#						
GETBUF	7-0#	20-37												
GETINX	7-0#	13-471*	13-471*	13-942*	13-942*	19-13	20-38							
GETSTS	13-71	13-113	13-158	13-218	13-260	13-304	13-359	13-401	13-447	13-537	13-580	13-625	13-685	13-728
	13-772	13-830	13-872	13-918	13-:09	13-:58	13-;02	13-;61	13-;91	13-<53	13-=46	13-=83	13->10	13->43
	13->70	13-?28	13-?76	13-@20	13-@86	13-A28	13-B23	13-C74	13-D16	13-D78	13-E94	13-F54	13-G18	13-G73
	13-H36	13-H80	13-I29	13-I91	13-J38	13-J85	13-K43	13-K85	13-L30	13-L88	13-M30	13-M75	13-N36	13-N78
	13-023	13-083	13-P36	13-P97	15-46	16-75	19-10#							
GNS	5-1	5-1	10-6	10-28	10-42	10-53	10-59	10-60	12-30	14-20	14-20	47-1	47-1	47-1
	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1
	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1					
GO	4-496#	13-50	13-100	13-145	13-197	13-247	13-291	13-339	13-388	13-434	13-514	13-567	13-612	13-662
	13-715	13-759	13-807	13-859	13-905	13-986	13-:45	13-:89	13-;38	13-;86	13-<08	13-<48	13-<68	13-=18
	13-=78	13-=97	13->38	13->57	13-?01	13-?63	13-@07	13-@57	13-A15	13-B10	13-C09	13-C43	13-D02	13-D64
	13-E34	13-E79	13-F38	13-F92	13-G05	13-G60	13-H15	13-H67	13-I16	13-I70	13-J22	13-J72	13-K22	13-K72
	13-L16	13-L67	13-M17	13-M62	13-N13	13-N65	13-010	13-062	13-P18	13-P28	13-P78	13-P89	15-126	15-181
	16-87	16-119	16-153	16-182	23-21	24-90	24-102	24-107	24-123	25-51	25-54	35-342		
GTSWR	10-28	47-1#												
HCE	4-574#	4-583	13-A51	13-A57	13-B44	13-B50	13-D28	13-D31	13-D90	13-D93	16-222	25-379	25-384	25-399
	25-401	25-414	25-429	35-348	35-378	35-381	35-391	35-394	51-77	51-78	51-81	51-85	51-86	
HCI	4-652#	13-D01	25-377	35-344										
HCRC	4-573#	4-583	16-222	25-386	35-348	35-352	35-355	35-391	35-394					
HELP	11-17	62-18#												
HT	4-484#	41-1	41-1	44-90										
IAE	4-571#	25-283	26-89	26-105	26-132	29-47	29-67	33-24	33-78	33-81	35-172	35-190	51-59	51-69
	51-77	51-78	51-81	51-82	51-85	51-86								
IBSAVE	43-1	43-1	43-1	43-1	43-1	43-1#	43-1*	43-1*	43-1*					
IDXMSK	4-710#	20-51	20-54	21-45	21-48									
IE	4-723#	4-756#												
ILF	4-581#	25-155	25-157	32-51	32-111	32-115	33-24	33-45	33-48	35-128	35-145	35-148	51-58	51-67
	51-68	51-70	51-71	51-72	51-73	51-74	51-75	51-76	51-79	51-80	51-83	51-84	51-87	51-88
ILF02	4-501#													
ILF24	4-511#													
ILF26	4-512#													
ILF30	4-516#													
ILF32	4-516#													
ILF34	4-516#													
ILF36	4-516#													
ILF40	4-516#													
ILF42	4-516#													
ILF44	4-516#													
ILF46	4-516#													
ILF54	4-519#													
ILF56	4-520#													
ILF64	4-523#													
ILF66	4-524#													
ILF74	4-527#													
ILF76	4-528#													
ILR	4-580#	32-51	32-97	32-101	33-232	33-236	33-239	35-128	35-131	35-134				
ILRG50	4-708#													
ILRG52	4-708#													
ILRG54	4-708#													
ILRG56	4-708#													
ILRG60	4-708#													
ILRG62	4-708#													
ILRG64	4-708#													

```
QUES     50-6#
R6        4-484#  10-23   10-23*  10-23*
R7        4-484#
RCLSTS   15-207   16-171  33-15#
RD        4-525#  13-145  13-612  13-<68  13-B10  13-D64  13-F38  13-G60  13-I16  13-L16  13-M62  13-O10  13-P28
RDCHR    11-12    11-65   11-82   45-1    47-1#
RDLIN    46-1     47-1#
RDOCT    11-35    11-49   47-1#
RDY       4-722#  23-21   23-22   24-86   24-89   31-31
READY    12-37#   14-24
RECAL     4-503#  15-181  16-153
RESREG   44-156   47-1#
RESVEC    4-484#
REX       4-624#
RG        4-622#
RGDTPT   53-3#
RH        4-526#  16-182
RIP       4-508#
RLEASE    4-505#
RMAS      4-694#
RMASI     7-0#    34-50   60-13
RMASO     7-0#
RMBA      4-766#  13-12*  13-13   13-62   13-105  13-150  13-209  13-252  13-296  13-350  13-393  13-439  13-526  13-572
         13-617   13-674  13-720  13-764  13-819  13-864  13-910  13-998  13-:50  13-:94  13-;50  13-<13  13-<74  13-=36
         13->02   13->62  13-?19  13-?68  13-@12  13-@77  13-A20  13-B15  13-C14  13-C63  13-D08  13-D70  13-E39  13-E87
         13-F46   13-G10  13-G65  13-H27  13-H72  13-I21  13-I82  13-J30  13-J77  13-K34  13-K77  13-L22  13-L79  13-M22
         13-M67   13-N25  13-N70  13-O15  13-O74  13-P13  13-P70  16-68
RMBAE     4-769#
RMBAEI    7-0#
RMBAEO    7-0#
RMBAI     7-0#    13-477  13-949  26-32   26-34   31-42   60-14
RMBAO     7-0#    13-47*  13-98*  13-144* 13-194* 13-245* 13-336* 13-387* 13-511* 13-565* 13-611* 13-659* 13-713* 13-804*
         13-858*  13-983*  13-:26* 13-;35* 13-<07* 13-<69* 13-=16* 13-=63* 13->99* 13-?46* 13-@54* 13-A13* 13-B08* 13-C03*
         13-C40*  13-D03*  13-D65* 13-E28* 13-E58* 13-F39* 13-F89* 13-H12* 13-H66* 13-I14* 13-I67* 13-J21* 13-K19* 13-K71*
         13-L17*  13-L64*  13-M16* 13-M61* 13-N10* 13-N63* 13-O09* 13-O59* 13-P00* 16-61*  16-253  16-256* 17-21   18-31
         26-26    26-30
RMCS1     4-690#   4-764#  10-89   13-9*   13-65   13-107  13-152  13-212  13-254  13-298  13-353  13-395  13-441  13-529
         13-574   13-619   13-677  13-722  13-766  13-822  13-866  13-912  13-:01  13-:52  13-:96  13-;53  13-<15  13-<76
         13- 39   13->04   13->64  13-?22  13-?70  13-@14  13-@80  13-A22  13-B17  13-C16  13-C66  13-D10  13-D72  13-E41
         13-E88   13-F48   13-G12  13-G67  13-H30  13-H74  13-I23  13-I85  13-J32  13-J79  13-K37  13-K79  13-L24  13-L82
         13-M24   13-M69   13-N28  13-N72  13-O17  13-O77  13-P20  13-P30  13-P80  13-P91  15-127  15-182  16-69   20-42
         21-36    23-20    28-28
RMCS1I    7-0#    16-209  24-65   24-67   24-69   24-86   24-88   24-91   24-102  24-106  24-108  24-120  24-122  24-124
         25-51    25-53    25-64  25-107  26-12   31-29   34-16   35-27   35-29   35-31   60-13
RMCS10    7-0#    13-50*  13-100* 13-145* 13-197* 13-247* 13-291* 13-339* 13-388* 13-434* 13-514* 13-567* 13-612* 13-662*
         13-715*  13-759*  13-807* 13-859* 13-905* 13-986* 13-:27* 13-:45* 13-:89* 13-;38* 13-;86* 13-<08* 13-<48* 13-<68*
         13-=18*  13-=66*  13-?78* 13-=97* 13->38* 13->57* 13-?01* 13-?44* 13-?63* 13-@07* 13-@57* 13-A15* 13-B10* 13-C04*
         13-C09*  13-C43*  13-D02* 13-D64* 13-E29* 13-E34* 13-E61* 13-E79* 13-F38* 13-F92* 13-G05* 13-G60* 13-H15* 13-H67*
         13-I16*  13-I70*  13-J22* 13-J72* 13-K22* 13-K72* 13-L16* 13-L67* 13-M17* 13-M62* 13-N13* 13-N65* 13-O10* 13-O62*
         13-P18*  13-P28*  13-P78* 13-P89* 15-126* 15-181* 16-87*  16-119* 16-153* 16-182* 16-304  17-26   25-28   26-48
         35-320   35-340
RMCS2     4-767#  10-84*  10-85*  10-87   12-58*  12-59*  13-14   13-15*  13-16   13-17*  20-41   21-35   28-27*  28-29
         30-16*   30-18*
RMCS2I    7-0#    13-458  13-463  13-465  13-929  13-933  13-935  24-47   24-72   24-74   24-75   25-99   25-170  25-271
         25-314   25-329  25-344  25-455  26-28   31-51   35-50   35-63   35-65   35-67   35-438  35-440  35-442  35-492
         35-494   35-496  60-13
RMCS20    7-0#
```

	13-F72	15-55	16-84	16-225	24-137	25-82	25-200	25-256	25-432	27-27	29-32	29-85	29-88	29-99
	29-127	29-130	29-179	29-208	31-114	33-31	33-92	33-99	33-101	33-103	33-116	33-118	33-120	33-130
	33-132	33-134	33-171	33-218	33-266	34-103	35-43	35-111	35-114	35-116	35-203	35 207	35-232	35-236
	35-238	35-240	35-250	35-254	35-256	35-266	35-268	35-270	35-284	35-466	35-468	35-470	35-513	35-546
	36-58	36-74	36-88	36-94	36-96	36-98	36-108	36-110	36-112	60-13				
RMER20	7-0#													
RMHR	4-701#													
RMHRI	7-0#													
RMHRO	7-0#													
RMLA	4-695#													
RMLAI	7-0#	60-16												
RMLAO	7-0#													
RMMR1	4-696#	13-E74	13-E82	13-F33	13-F42									
RMMR1I	7-0#	34-79	34-69	60-17										
RMMR1O	7-0#	13-E74*	13-E80*	13-F33*	13-F40*									
RMMR2	4-702#													
RMMR2I	7-0#	31-101	34-81	60-17										
RMMR2O	7-0#													
RMOF	4-699#	13-64	13-104	13-148	13-211	13-251	13-294	13-352	13-392	13-437	13-528	13-571	13-615	13-676
	13-719	13-762	13-821	13-863	13-908	13-:00	13-:49	13-:92	13-;52	13-<12	13-<72	13-=38	13->01	13->60
	13-?21	13-?67	13-@10	13-@79	13-A19	13-B13	13-C12	13-C65	13-D07	13-D68	13-E37	13-E85	13-F44	13-G09
	13-G63	13-H29	13-H71	13-I19	13-I84	13-J29	13-J75	13-K36	13-K76	13-L20	13-L81	13-M21	13-M65	13-N27
	13-N69	13-013	13-076	13-P12	13-P68	16-67								
RMOFI	7-0#	18-21	18-27	25-377	25-487	35-344	35-420	60-14	60-16					
RMOFO	7-0#	13-49*	13-196*	13-338*	13-513*	13-661*	13-806*	13-985*	13-;37*	13-=15*	13->98*	13-@56*	13-C02*	13-C42*
	13-D01*	13-E27*	13-E60*	13-F91*	13-H14*	13-I69*	13-K21*	13-L66*	13-N12*	13-061*	13-Q12	13-Q14*	16-60*	17-32
	29-58	35-183	60-11											
RMR	4-579#	32-51	32-83	32-87	33-232	33-250	33-253	35-128	35-159	35-162				
RMSN	4-698#													
RMSNI	7-0#	60-17												
RMSNO	7-0#													
RMWC	4-765#	13-10*	13-11	13-63	13-106	13-151	13-210	13-253	13-297	13-351	13-394	13-440	13-527	13-573
	13-618	13-675	13-721	13-765	13-820	13-865	13-911	13-999	13-:51	13-:95	13-;51	13-<14	13-<75	13-=37
	13->03	13->63	13-?20	13-?69	13-@13	13-@78	13-A21	13-B16	13-C15	13-C64	13-D09	13-D71	13-E40	13-E86
	13-F47	13-G11	13-G66	13-H28	13-H73	13-I22	13-I83	13-J31	13-J78	13-K35	13-K78	13-L23	13-L80	13-M23
	13-M68	13-N26	13-N71	13-016	13-075	13-P14	13-P71	16-66						
RMWCI	7-0#	18-72	26-10	26-23	26-44	26-115	60-14							
RMWCO	7-0#	13-48*	13-99*	13-195*	13-246*	13-337*	13-386*	13-512*	13-566*	13-660*	13-714*	13-805*	13-857*	13-984*
	13-:25*	13-;36*	13-<06*	13- 17*	13-=62*	13-?00*	13-?45*	13-@55*	13-A14*	13-B09*	13-C01*	13-C41*	13-E26*	13-E59*
	13-F90*	13-H13*	13-H65*	13-I15*	13-I68*	13-J20*	13-K20*	13-K70*	13-L65*	13-M15*	13-N11*	13-N64*	13-060*	13-099*
	16-59*	16-301	17-22	18-73	26-24	26-45								
RQA	4-662#	31-102	34-82											
RQB	4-663#	31-102	34-82											
RTC	4-507#													
SA1	4-544#													
SA16	4-540#													
SA2	4-543#													
SA4	4-542#													
SA8	4-541#													
SADMSK	4-548#	26-68												
SAVREG	44-13	47-1#												
SC	4-716#	24-90	24-107	24-123	31-30									
SC0	4-597#													
SC1	4-596#													
SC2	4-595#													
SC3	4-594#													
SC4	4-593#													

SW4	4-484#													
SW5	4-484#													
SW6	4-484#													
SW7	4-484#													
SW8	4-484#													
SW9	4-484#													
SWR	6-0#	10-23	10-23	10-23*	10-23*	10-23*	10-28	16-245	42-1	42-1	42-1	42-1	42-1	42-1
	42-1*	42-1*	42-1*	42-1*	43-1	43-1	43-1	43-1	43-1	44-14	45-1	45-1	45-1*	48-1
	48-1*													
SWREG	5-1#	10-23	10-28	45-1	45-1	45-1								
SYSTAT	10-76	50-23#												
TA1	4-537#													
TA16	4-533#	12-64												
TA2	4-536#	12-64												
TA4	4-535#	12-56												
TA8	4-534#													
TADMSK	4-547#	26-66												
TAG	4-664#													
TAGADR	5-9#	6-0												
TAP	4-641#													
TBITVE	4-484#													
TIMOUT	13-73	13-115	13-160	13-220	13-262	13-306	13-361	13-403	13-449	13-539	13-582	13-627	13-687	13-730
	13-774	13-832	13-874	13-920	13-:11	13-:60	13-;04	13-;63	13-;93	13-<21	13-<55	13-<82	13-=48	13-=85
	13->12	13->45	13->72	13-?30	13-?78	13-@22	13-@88	13-A30	13-B25	13-C20	13-C76	13-D18	13-D80	13-E45
	13-E96	13-F56	13-G31	13-G87	13-H38	13-H82	13-I31	13-I93	13-J40	13-J87	13-K45	13-K87	13-L32	13-L90
	13-M32	13-M77	13-N38	13-N80	13-025	13-085	13-P26	13-P38	13-P86	13-P99	15-137	15-192	16-96	16-128
	16-162	16-191	23-9#											
TKVEC	4-484#	45-1*	45-1*											
TPVEC	4-484#													
TRAPVE	4-484#	10-23*	10-23*											
TRE	4-717#	16-209	24-90	24-107	24-123	25-106	25-108	26-12						
TRTVEC	4-484#													
TST	4-665#	31-103	34-83											
TST1	13-2#	42-1												
TST10	13-969#	42-1												
TST11	13-;20#	42-1												
TST12	13-=05#	42-1												
TST13	13->88#	42-1												
TST14	13-@38#	42-1												
TST15	13-C24#	42-1												
TST16	13-E49#	42-1												
TST17	13-F81#	42-1												
TST2	13-33#	42-1												
TST20	13-G99#	42-1												
TST21	13-I54#	42-1												
TST22	13-K05#	42-1												
TST23	13-L50#	42-1												
TST24	13-M95#	42-1												
TST25	13-046#	42-1												
TST3	13-180#	42-1												
TST4	13-322#	42-1												
TST5	13-497#	42-1												
TST6	13-645#	42-1												
TST7	13-790#	42-1												
TSTNMB	44-44*	44-45*	44-47	44-159#										
TSTPRP	13-41	13-93	13-139	13-188	13-240	13-286	13-330	13-381	13-429	13-505	13-560	13-606	13-653	13-708
	13-754	13-798	13-852	13-900	13-977	13-:40	13-:84	13-;28	13-;83	13-<45	13-=29	13-=76	13->36	13-?12

```
$$CMRE   5-10#
$$CMTM   5-10#    6-0      6-0      6-0      6-0      6-0
$$ESCA   4-484#
$$NEWT   4-484#  13-2     13-33    13-180   13-322   13-497   13-645   13-790   13-969   13-;20   13-=05   13->88   13-@38   13-C24
        13-E49   13-F81   13-G99   13-I54   13-K05   13-L50   13-M95   13-046
$$SET    47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1     47-1#
$$SETM   10-23    10-23#
$$SKIP   4-484#
.$ACT1   4-476#   5-5
.$APTB   4-476#   6-0      6-0#
.$APTH   4-476#   5-8
.$APTY   4-476#  49-1
.$CATC   4-472#   5-1
.$CMTA   4-473#   5-10
.$EOP    4-473#  14-20
.$ERRO   4-473#  43-1
.$POWE   4-475#  48-1
.$RDDE   4-474#
.$RDOC   4-474#  46-1
.$READ   4-474#  45-1
.$SAVE   4-475#  37-1
.$SCOP   4-473#  42-1
.$SIZE   4-475#
.$TRAP   4-475#  47-1
.$TYPB   4-474#  38-1
.$TYPD   4-474#  39-1
.$TYPE   4-473#  41-1
.$TYPO   4-474#  40-1
.EQUAT   4-472#   4-484
.HEADE   4-472#   4-480
.SETUP   4-472#   4-775
.SWRHI   4-472#   4-481
.SWRLO   4-472#   4-481#   4-482
CALCLR   4-187#
CALSUB   4-201#  13-41    13-52    13-68    13-71    13-73    13-76    13-80    13-93    13-110   13-113   13-115   13-118   13-122
        13-124   13-126   13-139   13-155   13-158   13-160   13-163   13-167   13-169   13-171   13-175   13-188   13-199   13-215
        13-218   13-220   13-223   13-227   13-240   13-257   13-260   13-262   13-265   13-269   13-271   13-273   13-286   13-301
        13-304   13-306   13-309   13-313   13-315   13-317   13-330   13-341   13-356   13-359   13-361   13-364   13-368   13-381
        13-398   13-401   13-403   13-406   13-410   13-412   13-414   13-429   13-444   13-447   13-449   13-452   13-456   13-461
        13-489   13-505   13-516   13-534   13-537   13-539   13-542   13-546   13-560   13-577   13-580   13-582   13-585   13-589
        13-591   13-593   13-606   13-622   13-625   13-627   13-630   13-634   13-636   13-638   13-640   13-653   13-664   13-682
        13-685   13-687   13-690   13-694   13-708   13-725   13-728   13-730   13-733   13-737   13-739   13-741   13-754   13-769
        13-772   13-774   13-777   13-781   13-783   13-785   13-798   13-809   13-827   13-830   13-832   13-835   13-839   13-852
        13-869   13-872   13-874   13-877   13-881   13-883   13-885   13-900   13-915   13-918   13-920   13-923   13-927   13-931
        13-961   13-977   13-988   13-:06   13-:09   13-:11   13-:14   13-:18   13-:40   13-:55   13-:58   13-:60   13-:63   13-:67
        13-:69   13-:71   13-:84   13-:99   13-;02   13-;04   13-;07   13-;11   13-;13   13-;15   13-;28   13-;40   13-;58   13-:61
        13-;63   13-;66   13-;70   13-;83   13-;88   13-;91   13-;93   13-;96   13-<00   13-<18   13-<21   13-<24   13-<28   13-<30
        13-<32   13-<45   13-<50   13-<53   13-<55   13-<58   13-<62   13-<79   13-<82   13-<85   13-<89   13-<91   13-<93   13-<95
        13-=20   13-=29   13-=43   13-=46   13-=48   13-=51   13-=55   13-=76   13-=80   13-=83   13-=85   13-=88   13-=92   13->07
        13->10   13->12   13->15   13->19   13->21   13->23   13->36   13->40   13->43   13->45   13->48   13->52   13->67   13->70
        13->72   13->75   13->79   13->81   13->83   13-?03   13-?12   13-?25   13-?28   13-?30   13-?33   13-?37   13-?58   13-?73
        13-?76   13-?78   13-?81   13-?85   13-?87   13-?89   13-@02   13-@17   13-@20   13-@22   13-@25   13-@29   13-@31   13-@33
        13-@48   13-@59   13-@83   13-@86   13-@88   13-@91   13-@95   13-A08   13-A25   13-A28   13-A30   13-A33   13-A37   13-A54
        13-A70   13-A82   13-A91   13-B03   13-B20   13-B23   13-B25   13-B28   13-B32   13-B47   13-B63   13-B75   13-B84   13-C07
        13-C18   13-C34   13-C45   13-C71   13-C74   13-C76   13-C79   13-C83   13-C96   13-D13   13-D16   13-D18   13-D21   13-D25
        13-D44   13-D46   13-D59   13-D75   13-D78   13-D80   13-D83   13-D87   13-E06   13-E08   13-E10   13-E32   13-E43   13-E63
```

CROSS REFERENCE TABLE (CREF V01-05)

	22-4	22-5	23-9	24-141	25-128	28-20	30-12	31-52	34-51	34-52	37-1	39-1	46-1	48-1
	48-1	49-1	49-1	49-1										
PUTREG	4-442#	13-E74	13-F33	13-G00	13-G55									
REPORT	4-484#													
RGBFMC	4-69#	7-0	7-0											
SETPRI	4-484#	12-49	42-5	42-9	45-1									
SETTRA	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1	47-1#
SETUP	4-484#	10-23												
SKIP	4-484#													
SLASH	4-484#													
STARS	4-484#	5-5	5-8	5-8	5-8	6-0	6-0	6-0	13-2	13-2	13-33	13-33	13-35	13-87
	13-133	13-180	13-180	13-182	13-234	13-280	13-322	13-322	13-324	13-375	13-420	13-497	13-497	13-499
	13-553	13-600	13-645	13-645	13-647	13-701	13-748	13-790	13-790	13-792	13-846	13-891	13-969	13-969
	13-971	13-:33	13-:78	13-;20	13-;20	13-;22	13-;77	13-<39	13-=05	13-=05	13-=07	13-=70	13->30	13->88
	13->88	13->90	13-?52	13-?96	13-@38	13-@38	13-@40	13-A02	13-A97	13-B97	13-B99	13-C24	13-C24	13-C26
	13-C90	13-D53	13-E22	13-E24	13-E49	13-E49	13-E51	13-F23	13-F81	13-F81	13-F83	13-G45	13-G99	13-G99
	13-H02	13-H52	13-I02	13-I54	13-I54	13-I57	13-J07	13-J60	13-K05	13-K05	13-K07	13-K59	13-L05	13-L50
	13-L50	13-L52	13-M04	13-M50	13-M95	13-M95	13-M97	13-N52	13-N98	13-046	13-046	13-048	13-P01	13-P56
	14-20	15-57	15-70	15-84	15-107	15-139	15-161	15-194	16-213	16-290	16-294	25-26	25-34	25-119
	26-1	26-3	37-1	38-1	39-1	40-1	41-1	42-1	43-1	45-1	45-1	45-1	45-1	45-1
	46-1	47-1	48-1	48-1	49-1									
SWRSU	4-484#	10-23	10-23#											
TAGS	4-103#	6-0												
TRMTRP	47-1#													
TYPBIN	4-484#													
TYPDEC	4-484#	14-20	14-20											
TYPNAM	4-472#	4-484#	10-28											
TYPNUM	4-484#													
TYPOCS	4-484#	10-81	12-16	12-70	44-22	44-47	44-52	44-54						
TYPOCT	4-484#	11-33	45-1											
TYPTXT	4-484#	10-6	10-42	10-53	10-59	10-60	12-30	14-20	14-20					
XPER	4-12#.	8-3	8-6	8-9	8-12	8-15	8-18	8-21	8-24	8-27	8-30	8-33	8-36	8-39
	8-42	8-45	8-48	8-51	8-54	8-57	8-60	8-63	8-66	8-69	8-72	8-75	8-78	8-81
	8-84	8-87	8-90	8-93	8-96	8-99	8-102	8-105	8-108	8-111	8-114	8-117	8-120	8-123
	8-126	8-130	8-133	8-136	8-139	8-142	8-146	8-150	8-154	8-157	8-160	8-163	8-166	8-169
	8-172	8-175	8-178	8-182	8-185	8-188	8-191	8-194	8-197	8-200	8-203	8-206	8-209	8-212
	8-215	8-218	8-221	8-224	8-227	8-230	8-233	8-236	8-239	8-242	8-245	8-248	8-251	8-254
	8-257	8-260	8-263	8-266	8-269	8-272	8-275	8-278	8-281	8-284	8-287	8-290	8-293	8-296
	8-299	8-302	8-305	8-308	8-311	8-314	8-317	8-320	8-323	8-326	8-329	8-332	8-335	8-338
	8-341	8-344	8-347	8-351	8-354	8-357	8-360	8-363	8-366	8-369	8-372	8-375	8-378	8-381
	8-384	8-387	8-390	8-393	8-396	8-399	8-402	8-405	8-408	8-411	8-414	8-417	8-420	8-423
	8-426	8-429	8-432	8-435	8-438	8-441	8-444	8-447	8-450	8-453	8-456	8-459	8-462	8-465
	8-468	8-471	8-474	8-477	8-480	8-483	8-486	8-489	8-492	8-495	8-498	8-501	8-504	8-507
	8-510	8-513	8-516	8-519	8-522	8-525	8-528	8-531	8-534	8-537	8-540	8-543	8-546	8-549
	8-552	8-556	8-559	8-563	8-566	8-569	8-573	8-577	8-581	8-586	8-590	8-594	8-597	8-600
	8-603	8-606	8-609	8-612	8-615	8-618	8-621	8-624	8-627	8-630	8-633	8-636	8-639	8-642
	8-645	8-648	8-651	8-654	8-657	8-660	8-663	8-666	8-669	8-672	8-675	8-678	8-681	8-684
	8-687	8-690	8-693	8-696	8-699	8-702	8-705	8-708	8-711	8-714	8-717	8-720	8-723	