RM05/3/2

RM05/3/2 FCTNL TST2 AH-F925A-MC
CZRMNAO FICHE 1 OF 2

JUN 1980
COPYRIGHT© 1980
MADE IN USA

digital

RM05/3/2

RM05/3/2 FCTNL TST2
CZRMNA0

AH-F925A-MC
FICHE 2 OF 2

JUN 1980
COPYRIGHT© 1980
MADE IN USA

d i g i t a l

```
.REM    \
```

IDENTIFICATION

PRODUCT CODE: AC-F924A-MC

PRODUCT NAME: CZRMNA0 RM05/3/2 FCTNL TST 2

DATE CREATED: APRIL 1980

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1980 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
16K 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 DIAGNOSTIC, PART 1 & 2

RM05/3/2 FUNCTIONAL TEST, PART 1

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 STARTS AT LOCATION 200, WHICH PROVIDES WORST CASE TEST CONDITIONS IF RUNNING IN AN AUTOMATIC ENVIRONMENT. 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.

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 (Y/N) ?". IF THE OPERATOR RESPONDS WITH A "Y", THE PROGRAM WILL TYPE A BRIEF HELP MESSAGE WHICH WILL LIST SWITCH OPTIONS, ETC.

NOTE: THE FIRST QUESTION IS ONLY ASKED ON THE INITIAL PROGRAM START AND NOT ON SUBSEQUENT START-UP'S.

THE SECOND QUESTION TYPED IS, "CHANGE ADDRESSES (Y/N) ?". IF THE UNIBUS ADDRESS OF THE RH/RM IS NON STANDARD, THE OPERATOR SHOULD RESPOND WITH A "Y", THEN ANSWER SUBSEQUENT QUESTIONS TO SPECIFY THE UNIBUS ADDRESS, VECTOR ADDRESS AND INTERRUPT PRIORITY. IF THE OPERATOR RESPONSE IS A "N", THE PROGRAM SKIPS TO THE THIRD QUESTION.

THE THIRD 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 THE NUMBER(S) OF THE DRIVE(S) HE WANTS TESTED AND TERMINATE HIS INPUT WITH A "CARRIAGE RETURN".

NOTE: THE LONG VERSION OF THE THIRD QUESTION IS ONLY TYPED ON THE INITIAL PROGRAM START. ON ALL SUBSEQUENT STARTS, ONLY THE 'DRIVE(S):' PROMPT IS TYPED.

IF THE PROGRAM IS RESTARTED, THE FIRST QUESTION TYPED IS, 'USE SAME DEVICES (Y/N) ?'. IF THE OPERATOR TYPES "Y", THE TEST IS RESTARTED USING THE SAME DEVICES AS THE LAST TIME, OTHERWISE, THE TEST RESTARTS THE DIALOGUE AS IF THE PROGRAM WERE STARTED FOR THE FIRST TIME.

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 INCLUDES A MESSAGE AND AN ERROR SUMMARY.

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 BEING TESTED, 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.

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

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.

DEVICE AVAILABLE TEST

PURPOSE:

TO VERIFY THAT THE UNIT UNDER TEST IS AVAILABLE FOR TESTING, AND IS NOT LOCKED OR PROGRAMMED TO THE ALTERNATE PORT.

PROCEDURE:

THIS TEST SELECTS THE DEVICE AND READS CONTROL STATUS REGISTERS 1 AND 2 TO VERIFY THAT THE SELECTED DEVICE IS AVAILABLE FOR TESTING, AS INDICATED BY DVA STATUS, BIT 11 OF RMCS1 AND NED STATUS, BIT 12 OF RMCS2. THE RESULTS OF THE TEST CAN VARY AS FOLLOWS:

.NONEXISTANT DEVICE - THE DEVICE IS NONEXISTENT OR IS LOCKED ON THE ALTERNATE PORT AND IS THEREFORE NOT AVAILABLE FOR TESTING;

.DEVICE NOT AVAILABLE - THE DEVICE EXISTS BUT IS SEIZED BY THE ALTERNATE PORT AND IS NOT AVAILABLE FOR TESTING;

.BUS TIMEOUT - THE MASSBUS CONTROLLER FAILED TO DETECT A NONEXISTENT DEVICE;

.DEVICE AVAILABLE - THE DEVICE IS AVAILABLE FOR TESTING.

THE PROGRAM JUMPS TO THE SUBPASS HANDLER WHICH SELECTS THE NEXT DRIVE TO BE TESTED IF THE DEVICE IS NOT AVAILABLE.

DRIVE TYPE TEST

PURPOSE:

TO VERIFY THAT THE UNIT UNDER TEST IS AN RM05, RM03 OR RM02 SINGLE PORT OR DUAL PORT SUBSYSTEM.

PROCEDURE:

THIS TEST READS THE DRIVE TYPE REGISTER, RMDT, OF THE SELECTED DEVICE AND VERIFIES THAT THE DEVICE IS A SINGLE PORT OR DUAL PORT RM05/3/2 SUBSYSTEM. IF THE SELECTED DEVICE IS NOT AN RM05/3/2, THE PROGRAM JUMPS TO THE SUBPASS HANDLER WHICH WILL SELECT THE NEXT DEVICE FOR TESTING.

WRITE/READ HEADER AND DATA (FORMAT) TESTS

PURPOSE:

TO TEST WRITE HEADER AND DATA AND READ HEADER AND 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. FOLLOWING THAT, THE TEST PERFORMS ANY EXPLICIT SEEKS REQUIRED FOR THE CONDITIONS OF THE TEST. REGISTERS ARE PRESET AND THE WRITE HEADER AND 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 HEADER AND DATA PORTION OF THE TEST, VERIFYING THE SAME TYPE OF ERRORS AS IN THE WRITE COMMAND.

NOTE:
THAT 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 OF THE LAST TRACK. UNLESS SPECIFIED OTHERWISE, ALL TESTS ARE IN 16 BIT FORMAT.

FORMAT ZEROS TEST

THE TEST SEEKS TO CYLINDER 0, THEN WRITES HEADER AND DATA ON SECTOR 0 IN 18 BIT FORMAT. THE HEADER AND DATA FIELDS ARE ALL ZEROS, CAUSING THE DEVICE TO USE NORMAL WRITE GATE. THE HEADER AND DATA ARE READ AND COMPARED WITH THE WRITE BUFFER. THE INITIAL SEEK POSITIONS THE HEAD SUCH THAT THERE IS NO IMPLIED

SEEK. THE TEST IS REPEATED FOR 16 BIT FORMAT.

ZERO FILL TEST TEST

THE TEST EXECUTES A SEEK TO CYLINDER 0 TO INSURE THAT THERE IS NO HEAD MOTION DURING DATA TRANSFER. THIS IS FOLLOWED BY A WRITE HEADER AND DATA COMMAND WITH THE WORD COUNT EQUAL TO THE SIZE OF THE HEADER WHICH CAUSES THE RH TO ZERO FILL THE DATA FIELD. THE READ HEADER AND DATA COMMAND THAT FOLLOWS READS A FULL SECTOR AND VERIFIES THAT DATA WAS ZERO FILLED.

FORMAT CHECK ZEROS TEST

THE TEST WRITES HEADER AND AN ALL ZEROS DATA FIELD, THEN PERFORMS A WRITE CHECK HEADER AND DATA COMMAND AND VERIFIES THERE ARE NO ERRORS.

FORMAT CHECK ZEROS W/ WCE ERROR TEST

THE TEST WRITES HEADER AND AN ALL ZEROS DATA FIELD. AFTER COMPLEMENTING THE LAST WORD OF THE WRITE BUFFER, THE TEST PERFORMS A WRITE CHECK HEADER AND DATA COMMAND AND VERIFIES THAT THE CORRECT WRITE CHECK ERROR IS DETECTED.

FORMAT ONES TEST

THE TEST WRITES HEADER AND AN ALL ONES DATA FIELD, THEN READS THE HEADER AND DATA, VERIFYING THE READ BUFFER WITH THE WRITE BUFFER. THE ALL ONES FIELD IS A CONSTANT FREQUENCY, AND THE DRIVE SHOULD USE NORMAL WRITE GATE.

FORMAT CHECK ONES TEST

THE TEST FORMATS AN ALL ONES DATA FIELD, THE PERFORMS A WRITE CHECK HEADER AND DATA COMMAND, VERIFYING THAT THERE ARE NO ERRORS.

```
457
458
459
460
461                                 FORMAT CHECK ONES W/ WCE ERROR TEST
462
463                                      THE TEST FORMATS AN ALL ONES DATA FIELD, THEN COMPLEMENTS
464                                 THE LAST WORD OF THE WRITE BUFFER.  A WRITE CHECK HEADER AND DATA
465                                 COMMAND IS EXECUTED, AND THE TEST VERIFIES THAT THE CORRECT WRITE
466                                 CHECK ERROR IS DETECTED.
467
468
469
470
471
472                                 FORMAT MULTIPLE SECTORS TEST
473
474                                      THE TEST SEEKS TO CYLINDER 0 TO INSURE THERE IS NO HEAD
475                                 MOTION DURING DATA TRANSFER.  THE WRITE HEADER AND DATA COMMAND
476                                 FOLLOWS, WITH THE WORD COUNT EQUAL TO MULTIPLE SECTORS.  THE SAME
477                                 SECTORS ARE VERIFIED WITH A WRITE CHECK HEADER AND DATA COMMAND.
478
479
480
481
482
483                                 FORMAT WITH HEAD SWITCHING TEST
484
485                                      THE TEST SEEKS TO CYLINDER 0 TO INSURE THERE IS NO HEAD
486                                 MOTION DURING DATA TRANSFER.  THE WRITE HEADER AND DATA COMMAND
487                                 STARTS WITH CYLINDER 0, TRACK 0, SECTOR 31.  THE WORD COUNT IS
488                                 EQUAL TO MULTIPLE SECTORS WHICH CAUSES THE SUBSYSTEM TO SWITCH
489                                 FROM TRACK 0 TO TRACK 1 AFTER THE FIRST OF THE MULTIPLE SECTORS
490                                 ARE WRITTEN.  THE SAME SECTORS ARE VERIFIED WITH A WRITE CHECK
491                                 HEADER AND DATA COMMAND, USING THE SAME WORD COUNT AND STARTING
492                                 SECTOR.
493
494
495
496
497
498                                 FORMAT WITH MID-TRANSFER SEEK TEST
499
500                                      THIS TEST WRITES MULTIPLE SECTORS STARTING WITH CYLINDER 0,
501                                 LAST TRACK AND SECTOR 31., CAUSING A MIDTRANSFER SEEK AFTER THE
502                                 FIRST SECTOR IS WRITTEN. THE SAME SECTORS ARE VERIFIED WITH WRITE
503                                 CHECK HEADER AND DATA COMMAND.
504
505
506
507
508
509                                 FORMAT WITH IMPLIED SEEK TEST
510
511                                      THIS TEST SEEKS TO THE LAST CYLINDER PRIOR TO WRITING HEADER
512                                 AND DATA ON CYLINDER 0.  THE EXPLICIT SEEK INSURES THAT THERE
513                                 WILL BE MAXIMUM HEAD MOTION DURING THE IMPLIED SEEK OF THE WRITE
```

COMMAND. THE SAME OPERATION, INCLUDING THE EXPLICIT SEEK IS REPEATED FOR READ HEADER AN' DATA.

FORMAT EACH SECTOR ADDRESS TEST

HEADERS AND DATA OF EACH SECTOR ON CYLINDER 0, TRACK 0 ARE FORMATTED AND READ WITH THE PROGRAM VERIFYING HEADERS AND DATA.

FORMAT EACH TRACK ADDRESS TEST

THIS TEST FORMATS SECTOR 0 OF EACH TRACK ON CYLINDER 0 AND READS EACH SECTOR WITH THE PROGRAM VERIFYING HEADERS AND DATA.

FORMAT PRIME CYLINDERS TEST

THIS TEST FORMATS AND READS SECTOR 0, TRACK 0 ON EACH PRIME CYLINDER, I.E., CYLINDERS 1,2,4,8,...,512.

READ HEADER & DATA IN LAST SECTOR TEST

THIS TEST READS HEADER AND DATA ON THE LAST SECTOR OF THE, DISK, I.E., AND VERIFIES THAT "LBT" STATUS SETS.

READ HEADER & DATA W/ AOE ERROR TEST

THIS TEST READS MULTIPLE SECTORS STARTING WITH THE LAST SECTOR 31. AND VERIFIES THAT AOE STATUS SETS.

READ INVALID SECTOR ADDRESS TEST

THIS TEST USES AN ILLEGAL SECTOR ADDRESS AND VERIFIES THAT IAE STATUS SETS.

READ INVALID TRACK ADDRESS TEST

THIS TEST USES AN ILLEGAL TRACK ADDRESS AND VERIFIES THAT IAE STATUS SETS.

READ INVALID CYLINDER ADDRESS TEST

THIS TEST USES AN ILLEGAL CYLINDER ADDRESS AND VERIFIES THAT IAE STATUS SETS.

FORMAT AT OFFSET TEST

THE PROGRAM SETS OFFSET MODE AND EXECUTES A WRITE HEADER AND DATA COMMAND, VERIFYING THAT OFFSET MODE IS RESET BY THE WRITE COMMAND.

IVC FORMAT TEST

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

FORMAT ERROR TEST

A SINGLE SECTOR IS FORMATTED WITH THE OFFSET REGISTER SET FOR 16 BIT FORMAT AFTER WHICH THE SAME SECTOR IS READ IN 18 BIT FORMAT WITH THE PROGRAM VERIFYING THAT FORMAT ERROR STATUS IS SET. THE SAME PROCEDURE IS REPEATED WITH THE SECTOR WRITTEN IN 18 BIT FORMAT AND READ IN 16 BIT FORMAT.

FORMAT HCE TEST (1ST AND 2ND HEADER WORDS)

```
628                         THESE TWO TESTS WRITE AN  INCORRECT  HEADER  THEN  READ  THE
629                     HEADER AND VERIFY THAT THE CORRECT HEADER ERROR IS DETECTED.  THE
630                     TESTS SETUP THE CORRECT HEADER, THEN COMPLEMENT BIT 0 AND USE THE
631                     MODIFIED  BUFFER  TO  WRITE  THE HEADER.  THE PROCESS IS REPEATED
632                     UNTIL EACH BIT POSITION HAS BEEN SEPARATELY TESTED.
633
634
635
636
637
638                 \
```

```
  1                             ;PROGRAM REVISION #001
486
487                             .TITLE  CZRMNAO RM05/3/2 FCTNL TST 2
                                ;*COPYRIGHT (C) 1980
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY MIKE LEAVITT
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C4), 1980.
                                ;*
488                             .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>
489                             ;*         7            TN128
                                ;*         6            TN64
                                ;*         5            TN32
                                ;*         4            TN16
                                ;*         3            TN8
                                ;*         2            TN4
                                ;*         1            TN2
                                ;*         0            TN1
490
491                             .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
```

```
                000020          BIT04   = 20
                000010          BIT03   = 10
                000004          BIT02   = 4
                000002          BIT01   = 2
                000001          BIT00   = 1
                001000          BIT9=BIT09
                000400          BIT8=BIT08
                000200          BIT7=BIT07
                000100          BIT6=BIT06
                000040          BIT5=BIT05
                000020          BIT4=BIT04
                000010          BIT3=BIT03
                000004          BIT2=BIT02
                000002          BIT1=BIT01
                000001          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
492
493                             .SBTTL  RM REGISTER BIT DEFINITIONS
494
495                             ;*RMCS1 CONTROL STATUS REGISTER
496
497             004000          DVA     = BIT11                 ;DEVICE AVAILABLE-READ ONLY
498             000040          F4      = BIT05                 ;FUNCTION CODE
499             000020          F3      = BIT04                 ;FUNCTION CODE
500             000010          F2      = BIT03                 ;FUNCTION CODE
501             000004          F1      = BIT02                 ;FUNCTION CODE
502             000002          F0      = BIT01                 ;FUNCTION CODE
503             000001          GO      = BIT00                 ;GO BIT
504             000077          FNCMSK  = 000077                ;FUNCTION CODE MASK
505
506                             ;FUNCTION CODES (BITS 01-05 OF RMCS1)
507             000000          NOP     = 000000                ;NOP COMMAND
508             000002          ILF02   = 000002                ;ILLEGAL COMMAND
509             000004          SEEK    = 000004                ;SEEK COMMAND
510             000006          RECAL   = 000006                ;RECALIBRATE COMMAND
511             000010          DRVCLR  = 000010                ;DRIVE CLEAR COMMAND
512             000012          RLEASE  = 000012                ;RELEASE COMMAND
513             000014          OFFSET  = 000014                ;OFFSET COMMAND
514             000016          RTC     = 000016                ;RETURN TO CENTERLINE COMMAND
515             000020          RIP     = 000020                ;READ IN PRESET COMMAND
516             000022          PAKACK  = 000022                ;PACK ACKNOWLEDGE COMMAND
517             000022          PACACK  = PAKACK
518             000024          ILF24   = 000024                ;ILLEGAL COMMAND
519             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
```

```
                        
100000                  DCK     = BIT15                 ;DATA CHECK ERROR
040000                  UNS     = BIT14                 ;DRIVE UNSAFE
020000                  OPI     = BIT13                 ;OPERATION INCOMPLETE
010000                  DTE     = BIT12                 ;DRIVE TIMING ERROR
004000                  WLE     = BIT11                 ;WRITE LOCK ERROR
002000                  IAE     = BIT10                 ;INVALID ADDRESS ERROR
001000                  AOE     = BIT09                 ;ADDRESS OVERFLOW ERROR
000400                  HCRC    = BIT08                 ;HEADER CRC ERROR
000200                  HCE     = BIT07                 ;HEADER COMPARE ERROR
000100                  ECH     = BIT06                 ;ECC "HARD" ERROR
000040                  WCF     = BIT05                 ;WRITE CLOCK FAILURE
000020                  FER     = BIT04                 ;FORMAT ERROR
000010                  PAR     = BIT03                 ;PARITY ERROR
000004                  RMR     = BIT02                 ;REGISTER MODIFICATION REFUSED
000002                  ILR     = BIT01                 ;ILLEGAL REGISTER
000001                  ILF     = BIT00                 ;ILLEGAL FUNCTION

115760                  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

000377                  ATNMSK  = 377                   ;MASK FOR ATTENTION BITS

                        ;*RMLA  LOOK AHEAD REGISTER

002000                  SC4     = BIT10                 ;SECTOR COUNT = 16
001000                  SC3     = BIT09                 ;SECTOR COUNT = 8
000400                  SC2     = BIT08                 ;SECTOR COUNT = 4
000200                  SC1     = BIT07                 ;SECTOR COUNT = 2
000100                  SC0     = BIT06                 ;SECTOR COUNT = 1

003700                  SCTMSK  = 003700                ;SECTOR COUNT MASK

                        ;*RMMR1 MAINTENANCE REGISTER #1

                        ;WRITE ONLY BITS
100000                  DBCK    = BIT15                 ;DEBUG CLOCK
040000                  DBEN    = BIT14                 ;DEBUG CLOCK ENABLE
020000                  DEBL    = BIT13                 ;DIAGNOSTIC END OF BLOCK
010000                  DTO     = BIT12                 ;DIAGNOSTIC TIMEOUT
004000                  MCLK    = BIT11                 ;MAINTENANCE CLOCK
002000                  MRD     = BIT10                 ;READ DATA
001000                  MUR     = BIT09                 ;UNIT READY
000400                  MOC     = BIT08                 ;ON CYLINDER
000200                  MSER    = BIT07                 ;SEEK ERROR
000100                  MDF     = BIT06                 ;DRIVE FAULT
000040                  MS      = BIT05                 ;SECTOR PULSE
000010                  MWP     = BIT03                 ;WRITE PROTECT
000004                  MI      = BIT02                 ;INDEX PULSE
000002                  MSC     = BIT01                 ;SECTOR COMPARE
000001                  DMD     = BIT00                 ;DIAGNOSTIC MODE

                        ;READ ONLY BITS
100000                  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 ERROR
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
```

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

                        ;*RMCS2 RH CONTROL STATUS REGISTER #2

                        DLT     = BIT15                 ;DATA LATE-READ ONLY
                        WCE     = BIT14                 ;WRITE CHECK ERROR-READ ONLY
                        UPE     = BIT13                 ;UNIBUS PARITY ERROR
                        NED     = BIT12                 ;NONEXISTANT DRIVE-READ ONLY
                        NEM     = BIT11                 ;NONEXISTANT MEMORY-READ ONLY
                        PGE     = BIT10                 ;PROGRAM ERROR-READ ONLY
                        MXF     = BIT09                 ;MISSED TRANSFER
                        MDPE    = BIT08                 ;MASSBUS DATA BUS PARITY ERROR-READ ONLY
                        OR      = BIT07                 ;OUTPUT READY-READ ONLY
                        IR      = BIT06                 ;INPUT READY-READ ONLY
                        CLR     = BIT05                 ;CONTROLLER CLEAR
                        PAT     = BIT04                 ;PARITY TEST
                        BAI     = BIT03                 ;UNIBUS ADDRESS INCREMENT INHIBIT
                        U2      = BIT02                 ;UNIT SELECT
                        U1      = BIT01                 ;UNIT SELECT
                        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  RH11/RH70 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 REGISTER

                                    .SBTTL  STARTING ADDRESS(ES)

    000200      000137  005420              JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
785
786                                 .SBTTL  ACT11 HOOKS

                                    ;;**************************************************************
                                    ;HOOKS REQUIRED BY ACT11
                000204                          $SVPC=.                 ;SAVE PC
                000046                          .=46
    000046      032530                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                          .=52
    000052      000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                000204                          .=$SVPC                 ;; RESTORE PC
787
788             001100              .=1100
789                                 .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      000001              $TSTM:  .WORD   1       ;;RUN TIM OF LONGEST TEST
    001106      000002              $PASTM: .WORD   2       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
    001110      000002              $UNITM: .WORD   2       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDED UNIT
    001112      000042                      .WORD   $ETEND-$MAIL/2  ;;LENGTH MAILBOX-ETABLE(WORDS)
790             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
```

```
      0                         .SBTTL  USER DEFINED TAGS

    001326  000000              CTLFG:  .WORD   0       ;CONTAINS CONTROL-C FLAG
    001330  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.
    001332     000              LSTRK:  .BYTE   0       ;LO BYTE = 0
    001333     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

    001334                      GETBUF:

                                ;REGISTER INPUT BUFFER
    001334  000000              RMCS1I: .WORD   0       ;CONTROL, STATUS REGISTER #1
    001336  000000              RMWCI:  .WORD   0       ;WORD COUNT REGISTER
    001340  000000              RMBAI:  .WORD   0       ;BUS ADDRESS REGISTER
    001342  000000              RMDAI:  .WORD   0       ;DISK ADDRESS REGISTER
    001344  000000              RMCS2I: .WORD   0       ;CONTROL, STATUS REGISTER #2
    001346  000000              RMDSI:  .WORD   0       ;DRIVE STATUS REGISTER
    001350  000000              RMER1I: .WORD   0       ;ERROR REGISTER #1
    001352  000000              RMASI:  .WORD   0       ;ATTENTION SUMMARY REGISTER
    001354  000000              RMLAI:  .WORD   0       ;LOOK AHEAD REGISTER
    001356  000000              RMDBI:  .WORD   0       ;DATA BUFFER
    001360  000000              RMMR1I: .WORD   0       ;MAINTENANCE REGISTER #1
    001362  000000              RMDTI:  .WORD   0       ;DRIVE TYPE REGISTER
    001364  000000              RMSNI:  .WORD   0       ;SERIAL NUMBER REGISTER
    001366  000000              RMOFI:  .WORD   0       ;OFFSET REGISTER
    001370  000000              RMDCI:  .WORD   0       ;DESIRED CYLINDER REGISTER
    001372  000000              RMHRI:  .WORD   0       ;HOLDING REGISTER
    001374  000000              RMMR2I: .WORD   0       ;MAINTENANCE REGISTER #2
    001376  000000              RMER2I: .WORD   0       ;ERROR REGISTER #2
    001400  000000              RMEC1I: .WORD   0       ;ECC POSITION REGISTER
    001402  000000              RMEC2I: .WORD   0       ;ECC PATTERN REGISTER
    001404  000000              RMBAEI: .WORD   0       ;BUS ADDRESS EXTENSION REGISTER
    001406  000000              RMCS3I: .WORD   0       ;CONTROL, STATUS REGISTER #3

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

    001410                      PUTBUF:

                                ;REGISTER OUTPUT BUFFER
    001410  000000              RMCS1O: .WORD   0       ;CONTROL, STATUS REGISTER #1
    001412  000000              RMWCO:  .WORD   0       ;WORD COUNT REGISTER
    001414  000000              RMBAO:  .WORD   0       ;BUS ADDRESS REGISTER
    001416  000000              RMDAO:  .WORD   0       ;DISK ADDRESS REGISTER
    001420  000000              RMCS2O: .WORD   0       ;CONTROL, STATUS REGISTER #2
    001422  000000              RMDSO:  .WORD   0       ;DRIVE STATUS REGISTER
    001424  000000              RMER1O: .WORD   0       ;ERROR REGISTER #1
    001426  000000              RMASO:  .WORD   0       ;ATTENTION SUMMARY REGISTER
    001430  000000              RMLAO:  .WORD   0       ;LOOK AHEAD REGISTER
    001432  000000              RMDBO:  .WORD   0       ;DATA BUFFER
    001434  000000              RMMR1O: .WORD   0       ;MAINTENANCE REGISTER #1
    001436  000000              RMDTO:  .WORD   0       ;DRIVE TYPE REGISTER
```

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

001464  000000          TSTQUE: .WORD   0               ;CONTAINS DEVICE POINTER
                                .BLKW   8.              ;TEST QUE FOR DEVICES UNDER TEST
001506  000000                  .WORD   0               ;TABLE TERMINATOR GOES HERE WHEN
                                                        ;ALL 8. DEVICES ARE UNDER TEST.

                        ;MEDIA ENABLE IS SET IF THE BAD SECTOR FILES HAVE BEEN RECOVERED
                        ;FOR THE UNIT UNDER TEST, OTHERWISE IT IS ZERO.
001510  000000          MEDENB: .WORD   0               ;MEDIA ENABLE

                        ;LOCATIONS "ASNDC" AND "ASNDC" CONTAIN THE CYLINDER, TRACK AND SECTOR
                        ;ADDRESS ASSIGNED BY THE BAD SECTOR MODULE.
001512  000000          ASNDC:  .WORD   0               ;ASSIGNED DESIRED CYLINDER
001514  000000          ASNDA:  .WORD   0               ;ASSIGNED TRACK, AND SECTOR
001516  000000          CLKADR: .WORD   0               ;UNIBUS ADDRESS OF KW11 CLOCK
001520  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.
001522                  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.
001551                  PUTINX: .BLKB   23.             ;PUT INDEX TABLE

                        ;PUT TAGS HERE
```

```
      0                                 .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


    001600                              $ERRTB:
 1
 2
 3                                      ;ERROR  1       WRONG UNIT SELECTED
 4
    001600  065144                              EMT1
    001602  071236                              EHT1
    001604  071362                              EDT1
    001606  071452                              EFT1

 5
 6                                      ;ERROR  2       DEVICE WENT UNAVAILABLE
 7
    001610  065150                              EMT2
    001612  071236                              EHT1
    001614  071362                              EDT1
    001616  071452                              EFT1

 8
 9                                      ;ERROR  3       DEVICE WENT NONEXISTENT
10
    001620  065156                              EMT3
    001622  071236                              EHT1
    001624  071362                              EDT1
    001626  071452                              EFT1

11
12                                      ;ERROR  4       CONTROLLER NOT READY
13
    001630  065164                              EMT4
    001632  071236                              EHT1
    001634  071362                              EDT1
    001636  071452                              EFT1

14
15                                      ;ERROR  5       DRIVE NOT READY AND GO NOT RESET
16
    001640  065172                              EMT5
    001642  071236                              EHT1
    001644  071362                              EDT1
    001646  071452                              EFT1

17
```

```
                                        ;ERROR  6       UNEXPECTED VALUE FOR ''ATA'' STATUS

001650  065200                                  EMT6
001652  071236                                  EHT1
001654  071362                                  EDT1
001656  071452                                  EFT1

                                        ;ERROR  7       BUS TIMEOUT TRYING TO READ OR WRITE REGISTER

001660  065206                                  EMT7
001662  000000                                  0
001664  000000                                  0
001666  000000                                  0

                                        ;ERROR  10      DRIVE NOT READY BUT GO IS RESET

001670  065214                                  EMT10
001672  071236                                  EHT1
001674  071362                                  EDT1
001676  071452                                  EFT1

                                        ;ERROR  11      GO NOT RESET BUT DRIVE IS READY

001700  065220                                  EMT11
001702  071236                                  EHT1
001704  071362                                  EDT1
001706  071452                                  EFT1

                                        ;ERROR  12      INCORRECT FUNCTION CODE

001710  065224                                  EMT12
001712  071236                                  EHT1
001714  071362                                  EDT1
001716  071452                                  EFT1

                                        ;ERROR  13      PARITY ERROR READING REMOTE REGISTERS

001720  065232                                  EMT13
001722  071236                                  EHT1
001724  071362                                  EDT1
001726  071452                                  EFT1

                                        ;ERROR  14      TRANSFER ERROR IS INCORRECT

001730  065244                                  EMT14
001732  071236                                  EHT1
001734  071362                                  EDT1
001736  071452                                  EFT1

                                        ;ERROR  15      INCORRECT WORD COUNT
```

```
40
   001740  065252                          EMT15
   001742  071236                          EHT1
   001744  071362                          EDT1
   001746  071452                          EFT1

41
42                                 ;ERROR  16      INCORRECT BUS ADDRESS
43
   001750  065260                          EMT16
   001752  071236                          EHT1
   001754  071362                          EDT1
   001756  071452                          EFT1

44
45                                 ;ERROR  17      INCORRECT LBT STATUS
46
   001760  065270                          EMT17
   001762  071236                          EHT1
   001764  071362                          EDT1
   001766  071452                          EFT1

47
48                                 ;ERROR  20      INCORRECT AOE
49
   001770  065300                          EMT20
   001772  071236                          EHT1
   001774  071362                          EDT1
   001776  071452                          EFT1

50
51                                 ;ERROR  21      INCORRECT DISK ADDRESS
52
   002000  065310                          EMT21
   002002  071236                          EHT1
   002004  071362                          EDT1
   002006  071452                          EFT1

53
54                                 ;ERROR  22      INCORRECT CYLINDER ADDRESS
55
   002010  065320                          EMT22
   002012  071236                          EHT1
   002014  071362                          EDT1
   002016  071452                          EFT1

56
57                                 ;ERROR  23      INCORRECT WLE STATUS
58
   002020  065330                          EMT23
   002022  071236                          EHT1
   002024  071362                          EDT1
   002026  071452                          EFT1

59
60                                 ;ERROR  24      INCORRECT UPE STATUS
61
```

```
002030  065340                          EMT24
002032  071236                          EHT1
002034  071362                          EDT1
002036  071452                          EFT1

62
63                              ;ERROR  25      INCORRECT WCF STATUS
64
002040  065350                          EMT25
002042  071236                          EHT1
002044  071362                          EDT1
002046  071452                          EFT1

65
66                              ;ERROR  26      INCORRECT WCE STATUS
67
002050  065360                          EMT26
002052  071236                          EHT1
002054  071362                          EDT1
002056  071452                          EFT1

68
69                              ;ERROR  27      INCORRECT MDPE STATUS
70
002060  065370                          EMT27
002062  071236                          EHT1
002064  071362                          EDT1
002066  071452                          EFT1

71
72                              ;ERROR  30      INCORRECT DCK STATUS
73
002070  065400                          EMT30
002072  071236                          EHT1
002074  071362                          EDT1
002076  071452                          EFT1

74
75                              ;ERROR  31      INCORRECT ECH STATUS
76
002100  065410                          EMT31
002102  071236                          EHT1
002104  071362                          EDT1
002106  071452                          EFT1

77
78                              ;ERROR  32      DLT SHOULD NOT BE SET
79
002110  065420                          EMT32
002112  071236                          EHT1
002114  071362                          EDT1
002116  071452                          EFT1

80
81                              ;ERROR  33      MXF SHOULD NOT BE SET
82
002120  065430                          EMT33
```

```
002122  071236                          EHT1
002124  071362                          EDT1
002126  071452                          EFT1

 83
 84                             ;ERROR  34      DTE SHOULD NOT BE SET
 85
002130  065440                          EMT34
002132  071236                          EHT1
002134  071362                          EDT1
002136  071452                          EFT1

 86
 87                             ;ERROR  35      INCORRECT HCRC STATUS
 88
002140  065450                          EMT35
002142  071236                          EHT1
002144  071362                          EDT1
002146  071452                          EFT1

 89
 90                             ;ERROR  36      INCORRECT HCE STATUS
 91
002150  065460                          EMT36
002152  071236                          EHT1
002154  071362                          EDT1
002156  071452                          EFT1

 92
 93                             ;ERROR  37      INCORRECT FER STATUS
 94
002160  065470                          EMT37
002162  071236                          EHT1
002164  071362                          EDT1
002166  071452                          EFT1

 95
 96                             ;ERROR  40      DPE SHOULD NOT BE SET (NOT A DATA COMMAND)
 97
002170  065500                          EMT40
002172  071236                          EHT1
002174  071362                          EDT1
002176  071452                          EFT1

 98
 99                             ;ERROR  41      LOST "MOL" DURING PACK ACKNOWLEDGE
100
002200  065506                          EMT41
002202  071236                          EHT1
002204  071362                          EDT1
002206  071452                          EFT1

101
102                            ;ERROR  42      UNSAFE ERROR DURING PACK ACKNOWLEDGE
103
002210  065516                          EMT42
002212  071236                          EHT1
```

```
002214  071362                          EDT1
002216  071452                          EFT1

104
105                             ;ERROR  43      "OPI" ERROR DURING PACK ACKNOWLEDGE
106
002220  065530                          EMT43
002222  071236                          EHT1
002224  071362                          EDT1
002226  071452                          EFT1

107
108                             ;ERROR  44      "RMR" ERROR DURING PACK ACKNOWLEDGE
109
002230  065540                          EMT44
002232  071236                          EHT1
002234  071362                          EDT1
002236  071452                          EFT1

110
111                             ;ERROR  45      "ILR" ERROR DURING PACK ACKNOWLEDGE
112
002240  065550                          EMT45
002242  071236                          EHT1
002244  071362                          EDT1
002246  071452                          EFT1

113
114                             ;ERROR  46      "ILF" ERROR DURING PACK ACKNOWLEDGE
115
002250  065560                          EMT46
002252  071236                          EHT1
002254  071362                          EDT1
002256  071452                          EFT1

116
117                              ERROR  47      COMPOSITE ERROR STATUS IS INCORRECT
118
002260  065570                          EMT47
002262  071236                          EHT1
002264  071362                          EDT1
002266  071452                          EFT1

119
120                             ;ERROR  50      PARITY ERROR WRITING REMOTE REGISTERS
121
002270  065576                          EMT50
002272  071236                          EHT1
002274  071362                          EDT1
002276  071452                          EFT1

122
123                             ;ERROR  51      INCORRECT IAE STATUS DURING SEEK COMMAND
124
002300  065606                          EMT51
002302  071236                          EHT1
002304  071362                          EDT1
```

```
002306  071452                          EFT1

125
126                             ;ERROR  52      OPI ERROR DURING SEEK - MEDIUM IS NOT ON LINE
127
002310  065620                          EMT52
002312  071236                          EHT1
002314  071362                          EDT1
002316  071452                          EFT1

128
129                             ;ERROR  53      OPI ERROR DURING SEEK - MEDIUM IS ON LINE, ASSUME
130                             ;               ON CYLINDER LATCH DIDN'T RESET
131
002320  065636                          EMT53
002322  071236                          EHT1
002324  071362                          EDT1
002326  071452                          EFT1

132
133                             ;ERROR  54      SEEK INCOMPLETE ERROR DURING SEEK COMMAND
134
002330  065654                          EMT54
002332  071236                          EHT1
002334  071362                          EDT1
002336  071452                          EFT1

135
136                             ;ERROR  55      DEVICE CHECK DURING SEEK COMMAND
137
002340  065664                          EMT55
002342  071236                          EHT1
002344  071362                          EDT1
002346  071452                          EFT1

138
139                             ;ERROR  56      PIP IS STILL SET AFTER SEEK - SKI IS RESET
140
002350  065676                          EMT56
002352  071236                          EHT1
002354  071362                          EDT1
002356  071452                          EFT1

141
142                             ;ERROR  57      ATA DID NOT SET DURING SEEK COMMAND
143
002360  065714                          EMT57
002362  071236                          EHT1
002364  071362                          EDT1
002366  071452                          EFT1

144
145                             ;ERROR  60      IVC ERROR DURING SEEK COMMAND - LOST
146                             ;               VOLUME VALID
147
002370  065724                          EMT60
002372  071236                          EHT1
```

```
        002374  071362                          EDT1
        002376  071452                          EFT1

    148
    149                                 ;ERROR  61      ERRONEOUS IVC ERROR DURING SEEK COMMAND -
    150                                 ;               VOLUME VALID IS STIL SET
    151
        002400  065742                          EMT61
        002402  071236                          EHT1
        002404  071362                          EDT1
        002406  071452                          EFT1

    152
    153                                 ;ERROR  62      MOL IS ZERO, BUT OPI WAS NOT
    154                                 ;               REPORTED DURING SEEK COMMAND
    155
        002410  065762                          EMT62
        002412  071236                          EHT1
        002414  071362                          EDT1
        002416  071452                          EFT1

    156
    157                                 ;ERROR  63      UNUSED
    158
        002420  000000                          0
        002422  000000                          0
        002424  000000                          0
        002426  000000                          0

    159
    160                                 ;ERROR  64      DRIVE DID NOT DETECT "IVC" ERROR DURING SEEK
    161
        002430  066000                          EMT64
        002432  071236                          EHT1
        002434  071362                          EDT1
        002436  071452                          EFT1

    162
    163                                 ;ERROR  65      DRIVE EXECUTED A SEEK WITH ERROR SET
    164
        002440  066020                          EMT65
        002442  071236                          EHT1
        002444  071362                          EDT1
        002446  071452                          EFT1

    165
    166                                 ;ERROR  66      UNEXPECTED ERROR SET IN RMER1
    167
        002450  066040                          EMT66
        002452  071236                          EHT1
        002454  071362                          EDT1
        002456  071452                          EFT1

    168
    169                                 ;ERROR  67      UNEXPECTED ERROR SET IN RMER2
    170
        002460  066052                          EMT67
```

```
002462  071236                          EHT1
002464  071362                          EDT1
002466  071452                          EFT1

171
172                             ;ERROR  70      ERRONEOUS ''IAE'' ERROR DURING RECALIBRATE
173
002470  066064                          EMT70
002472  071236                          EHT1
002474  071362                          EDT1
002476  071452                          EFT1

174
175                             ;ERROR  71      ''ILF'' ERROR DURING RECALIBRATE
176
002500  066074                          EMT71
002502  071236                          EHT1
002504  071362                          EDT1
002506  071452                          EFT1

177
178                             ;ERROR  72      ''OPI'' ERROR DURING RECALIBRATE DUE TO ''MOL'' - 0
179
002510  066104                          EMT72
002512  071236                          EHT1
002514  071362                          EDT1
002516  071452                          EFT1

180
181                             ;ERROR  73      ''OPI'' ERROR DURING RECALIBRATE BECAUSE ON
182                             ;               CYLINDER DIDNT DROP
183
002520  066122                          EMT73
002522  071236                          EHT1
002524  071362                          EDT1
002526  071452                          EFT1

184
185                             ;ERROR  74      ''IVC'' ERROR DURING RECALIBRATE - ''VV'' = 0
186
002530  066140                          EMT74
002532  071236                          EHT1
002534  071362                          EDT1
002536  071452                          EFT1

187
188                             ;ERROR  75      ERRONEOUS ''IVC'' ERROR DURING RECALIBRATE - ''VV'' - 1
189
002540  066150                          EMT75
002542  071236                          EHT1
002544  071362                          EDT1
002546  071452                          EFT1

190
191                             ;ERROR  76      ''SKI'' ERROR DURING RECALIBRATE
192
002550  066170                          EMT76
```

```
002552  071236                          EHT1
002554  071362                          EDT1
002556  071452                          EFT1

193
194                             ;ERROR  77      "DVC" OCCURRED DURING RECALIBRATE
195
002560  066200                          EMT77
002562  071236                          EHT1
002564  071362                          EDT1
002566  071452                          EFT1

196
197                             ;ERROR  100     LOST "MOL" DURING RECALIBRATE - "OPI" = 0
198
002570  066212                          EMT100
002572  071236                          EHT1
002574  071362                          EDT1
002576  071452                          EFT1

199
200                             ;ERROR  101     LOST "VV" DURING RECALIBRATE - "IVC" = 0
201
002600  066230                          EMT101
002602  071236                          EHT1
002604  071362                          EDT1
002606  071452                          EFT1

202
203                             ;ERROR  102     "ATA" DID NOT SET DURING RECALIBRATE
204
002610  066246                          EMT102
002612  071236                          EHT1
002614  071362                          EDT1
002616  071452                          EFT1

205
206                             ;ERROR  103     "OM" DID NOT RESET DURING RECALIBRATE
207
002620  066256                          EMT103
002622  071236                          EHT1
002624  071362                          EDT1
002626  071452                          EFT1

208
209                             ;ERROR  104     "PIP" IS STIL SET AFTER RECALIBRATE
210
002630  066270                          EMT104
002632  071236                          EHT1
002634  071362                          EDT1
002636  071452                          EFT1

211
212                             ;ERROR  105     UNEXPECTED "ILR" ERROR DURING RECALIBRATE
213
002640  066306                          EMT105
002642  071236                          EHT1
```

```
002644  071362                          EDT1
002646  071452                          EFT1

214
215                             ;ERROR  106     UNEXPECTED ''RMR'' ERROR DURING RECALIBRATE
216
002650  066316                          EMT106
002652  071236                          EHT1
002654  071362                          EDT1
002656  071452                          EFT1

217
218                             ;ERROR  107     ''UNS'' ERROR DURING RECALIBRATE - AC POWER IS LOW
219
002660  066326                          EMT107
002662  071236                          EHT1
002664  071362                          EDT1
002666  071452                          EFT1

220
221                             ;ERROR  110     CANNOT ACCESS MASSBUS CONTROLLER VIA UNIBUS
222
002670  066346                          EMT110
002672  071260                          EHT110
002674  071400                          EDT110
002676  071470                          EFT110

223
224                             ;ERROR  111     NONEXISTENT DEVICE
225
002700  066360                          EMT111
002702  071264                          EHT111
002704  071402                          EDT111
002706  071472                          EFT111

226
227                             ;ERROR  112     DEVICE NOT AVAILABLE
228
002710  066366                          EMT112
002712  071264                          EHT111
002714  071402                          EDT111
002716  071472                          EFT111

229
230                             ;ERROR  113     BUS TIMEOUT-NED STATUS FAILURE
231
002720  066374                          EMT113
002722  000000                          0
002724  000000                          0
002726  000000                          0

232
233                             ;ERROR  114     DEVICE NOT AN RM05/3/2
234
002730  066410                          EMT114
002732  071270                          EHT114
002734  071404                          EDT114
```

```
        002736  071474                          EFT114

    235
    236                             ;ERROR  115     RMCS1 NOT INITIALIZED BY UNIBUS
    237
        002740  066416                          EMT115
        002742  071236                          EHT1
        002744  071362                          EDT1
        002746  071452                          EFT1

    238
    239                             ;ERROR  116     RMBA NOT INITIALIZED BY UNIBUS
    240
        002750  066426                          EMT116
        002752  071236                          EHT1
        002754  071362                          EDT1
        002756  071452                          EFT1

    241
    242                             ;ERROR  117     RMCS2 NOT INITIALIZED BY UNIBUS
    243
        002760  066436                          EMT117
        002762  071236                          EHT1
        002764  071362                          EDT1
        002766  071452                          EFT1

    244
    245                             ;ERROR  120     RMER1 NOT INITIALIZED BY UNIBUS
    246
        002770  066446                          EMT120
        002772  071236                          EHT1
        002774  071362                          EDT1
        002776  071452                          EFT1

    247
    248                             ;ERROR  121     RMAS NOT INITIALIZED BY UNIBUS
    249
        003000  066456                          EMT121
        003002  071236                          EHT1
        003004  071362                          EDT1
        003006  071452                          EFT1

    250
    251                             ;ERROR  122     RMMR1 NOT INITIALIZED BY UNIBUS
    252
        003010  066466                          EMT122
        003012  071236                          EHT1
        003014  071362                          EDT1
        003016  071452                          EFT1

    253
    254                             ;ERROR  123     RMDS NOT INITIALIZED BY UNIBUS
    255
        003020  066476                          EMT123
        003022  071236                          EHT1
        003024  071362                          EDT1
        003026  071452                          EFT1
```

```
256
257                                     ;ERROR  124     RMEC2 NOT INITIALIZED BY UNIBUS
258
    003030  066506                              EMT124
    003032  071236                              EHT1
    003034  071362                              EDT1
    003036  071452                              EFT1

259
260                                     ;ERROR  125     RMMR2 NOT INITIALIZED BY UNIBUS
261
    003040  066516                              EMT125
    003042  071236                              EHT1
    003044  071362                              EDT1
    003046  071452                              EFT1

262
263                                     ;ERROR  126     RMCS1 NOT CLEARED BY CONTROLLER CLEAR
264
    003050  066526                              EMT126
    003052  071236                              EHT1
    003054  071362                              EDT1
    003056  071452                              EFT1

265
266                                     ;ERROR  127     RMBA NOT CLEARED BY CONTROLLER CLEAR
267
    003060  066540                              EMT127
    003062  071236                              EHT1
    003064  071362                              EDT1
    003066  071452                              EFT1

268
269                                     ;ERROR  130     RMCS2 NOT CLEARED BY CONTROLLER CLEAR
270
    003070  066552                              EMT130
    003072  071236                              EHT1
    003074  071362                              EDT1
    003076  071452                              EFT1

271
272                                     ;ERROR  131     RMER1 NOT CLEARED BY CONTROLLER CLEAR
273
    003100  066564                              EMT131
    003102  071236                              EHT1
    003104  071362                              EDT1
    003106  071452                              EFT1

274
275                                     ;ERROR  132     RMAS NOT CLEARED BY CONTROLLER CLEAR
276
    003110  066576                              EMT132
    003112  071236                              EHT1
    003114  071362                              EDT1
    003116  071452                              EFT1
```

```
277
278                                 ;ERROR  133     RMMR1 NOT CLEARED BY CONTROLLER CLEAR
279
    003120  066610                          EMT133
    003122  071236                          EHT1
    003124  071362                          EDT1
    003126  071452                          EFT1

280
281                                 ;ERROR  134     RMDS NOT CLEARED BY CONTROLLER CLEAR
282
    003130  066622                          EMT134
    003132  071236                          EHT1
    003134  071362                          EDT1
    003136  071452                          EFT1

283
284                                 ;ERROR  135     RMEC2 NOT CLEARED BY CONTROLLER CLEAR
285
    003140  066634                          EMT135
    003142  071236                          EHT1
    003144  071362                          EDT1
    003146  071452                          EFT1

286
287                                 ;ERROR  136     RMMR2 NOT CLEARED BY CONTROLLER CLEAR
288
    003150  066646                          EMT136
    003152  071236                          EHT1
    003154  071362                          EDT1
    003156  071452                          EFT1

289
290                                 ;ERROR  137     RMCS1 NOT CLEARED BY ERROR CLEAR
291
    003160  066660                          EMT137
    003162  071236                          EHT1
    003164  071362                          EDT1
    003166  071452                          EFT1

292
293                                 ;ERROR  140     RMCS2 NOT CLEARED BY ERROR CLEAR
294
    003170  066670                          EMT140
    003172  071236                          EHT1
    003174  071362                          EDT1
    003176  071452                          EFT1

295
296                                 ;ERROR  141     RMCS1 NOT CLEARED BY DRIVE CLEAR
297
    003200  066700                          EMT141
    003202  071236                          EHT1
    003204  071362                          EDT1
    003206  071452                          EFT1

298
```

```
299                                     ;ERROR  142     RMDS NOT CLEARED BY DRIVE CLEAR
300
    003210  066710                              EMT142
    003212  071236                              EHT1
    003214  071362                              EDT1
    003216  071452                              EFT1

301
302                                     ;ERROR  143     RMER1 NOT CLEARED BY DRIVE CLEAR
303
    003220  066720                              EMT143
    003222  071236                              EHT1
    003224  071362                              EDT1
    003226  071452                              EFT1

304
305                                     ;ERROR  144     RMAS NOT CLEARED BY DRIVE CLEAR
306
    003230  066730                              EMT144
    003232  071236                              EHT1
    003234  071362                              EDT1
    003236  071452                              EFT1

307
308                                     ;ERROR  145     RMMR1  NOT CLEARED BY DRIVE CLEAR
309
    003240  066740                              EMT145
    003242  071236                              EHT1
    003244  071362                              EDT1
    003246  071452                              EFT1

310
311                                     ;ERROR  146     RMMR2 NOT CLEARED BY DRIVE CLEAR
312
    003250  066750                              EMT146
    003252  071236                              EHT1
    003254  071362                              EDT1
    003256  071452                              EFT1

313
314                                     ;ERROR  147     RMER2 NOT CLEARED BY DRIVE CLEAR
315
    003260  066760                              EMT147
    003262  071236                              EHT1
    003264  071362                              EDT1
    003266  071452                              EFT1

316
317                                     ;ERROR  150     RMEC2 NOT CLEARED BY DRIVE CLEAR
318
    003270  066770                              EMT150
    003272  071236                              EHT1
    003274  071362                              EDT1
    003276  071452                              EFT1

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

```
321
    003300  067000                          EMT151
    003302  071236                          EHT1
    003304  071362                          EDT1
    003306  071452                          EFT1

322
323                                 ;ERROR  152     DRIVE FAULT
324
    003310  067012                          EMT152
    003312  071236                          EHT1
    003314  071362                          EDT1
    003316  071452                          EFT1

325
326                                 ;ERROR  153     UNSAFE SHOULD BE SET BECAUSE DVC IS SET
327
    003320  067024                          EMT153
    003322  071236                          EHT1
    003324  071362                          EDT1
    003326  071452                          EFT1

328
329                                 ;ERROR  154     UNSAFE SHOULD NOT BE SET, AC IS LOW
330
    003330  067042                          EMT154
    003332  071236                          EHT1
    003334  071362                          EDT1
    003336  071452                          EFT1

331
332                                 ;ERROR  155     VOLUME VALID NOT SET BY PACK ACK
333
    003340  067060                          EMT155
    003342  071236                          EHT1
    003344  071362                          EDT1
    003346  071452                          EFT1

334
335                                 ;ERROR  156     OFFSET MODE NOT SET BY OFFSET COMMAND
336
    003350  067072                          EMT156
    003352  071236                          EHT1
    003354  071362                          EDT1
    003356  071452                          EFT1

337
338                                 ;ERROR  157     OFFSET MODE NOT RESET BY RTC COMMAND
339
    003360  067104                          EMT157
    003362  071236                          EHT1
    003364  071362                          EDT1
    003366  071452                          EFT1

340
341                                 ;ERROR  160     RMOF NOT RESET BY RIP COMMAND
342
```

```
003370  067116                          EMT160
003372  071236                          EHT1
003374  071362                          EDT1
003376  071452                          EFT1

343
344                             ;ERROR  161     RMDA NOT RESET BY RIP COMMAND
345
003400  067126                          EMT161
003402  071236                          EHT1
003404  071362                          EDT1
003406  071452                          EFT1

346
347                             ;ERROR  162     RMDC NOT RESET BY RIP COMMAND
348
003410  067140                          EMT162
003412  071236                          EHT1
003414  071362                          EDT1
003416  071452                          EFT1

349
350                             ;ERROR 163      DATA WAS ECC CORRECTED BUT DOES NOT COMPARE WITH
351                             :               WRITE BUFFER
352
003420  071032                          EMT336
003422  071320                          EHT336
003424  071416                          EDT336
003426  071506                          EFT336

353
354                             ;ERROR  164     OPI SHOULD NOT BE SET
355
003430  067162                          EMT164
003432  071236                          EHT1
003434  071362                          EDT1
003436  071452                          EFT1

356
357                             ;ERROR  165     IVC SHOULD NOT BE SET
358
003440  067170                          EMT165
003442  071236                          EHT1
003444  071362                          EDT1
003446  071452                          EFT1

359
360                             ;ERROR  166     IAE SHOULD NOT BE SET
361
003450  067176                          EMT166
003452  071236                          EHT1
003454  071362                          EDT1
003456  071452                          EFT1

362
363                             ;ERROR  167     NEM SHOULD NOT BE SET
364
```

```
003460  067204                          EMT167
003462  071236                          EHT1
003464  071362                          EDT1
003466  071452                          EFT1

365
366                             ;ERROR 170      UNUSED
367
003470  000000                          0
003472  000000                          0
003474  000000                          0
003476  000000                          0

368
369                             ;ERROR  171     "ATA" NOT SET DURING RETURN TO CENTERLINE
370
003500  067222                          EMT171
003502  071236                          EHT1
003504  071362                          EDT1
003506  071452                          EFT1

371
372                             ;ERROR  172     "ATA" NOT SET BY OFFSET COMMAND
373
003510  067232                          EMT172
003512  071236                          EHT1
003514  071362                          EDT1
003516  071452                          EFT1

374
375                             ;ERROR  173     RMER2 NOT INITIALIZED BY UNIBUS INIT
376
003520  067242                          EMT173
003522  071236                          EHT1
003524  071362                          EDT1
003526  071452                          EFT1

377
378                             ;ERROR  174     RMER2 NOT INITIALIZED BY CONTROLLER CLEAR
379
003530  067252                          EMT174
003532  071236                          EHT1
003534  071362                          EDT1
003536  071452                          EFT1

380
381                             ;ERROR  175     SELECTED DEVICE IS IN WRITE PROTECT
382
003540  067264                          EMT175
003542  071236                          EHT1
003544  071362                          EDT1
003546  071452                          EFT1

383
384                             ;ERROR  176     CANNOT SET DIAGNOSTIC MODE
385
003550  067272                          EMT176
```

```
003552  071236                          EMT1
003554  071362                          EDT1
003556  071452                          EFT1

386
387                             ;ERROR  177     INCORRECT "MOL" STATUS DURING DIAGNOSTIC MODE
388
003560  067300                          EMT177
003562  071236                          EMT1
003564  071362                          EDT1
003566  071452                          EFT1

389
390                             ;ERROR  200     INCORRECT "PIP" STATUS DURING DIAGNOSTIC MODE
391
003570  067312                          EMT200
003572  071236                          EMT1
003574  071362                          EDT1
003576  071452                          EFT1

392
393                             ;ERROR  201     INCORRECT "WRL" STATUS DURING DIAGNOSTIC MODE
394
003600  067324                          EMT201
003602  071236                          EMT1
003604  071362                          EDT1
003606  071452                          EFT1

395
396                             ;ERROR  202     INCORRECT "SKI" STATUS DURING DIAGNOSTIC MODE
397
003610  067336                          EMT202
003612  071236                          EMT1
003614  071362                          EDT1
003616  071452                          EFT1

398
399                             ;ERROR  203     INCORRECT "DVC" STATUS DURING DIAGNOSTIC MODE
400
003620  067350                          EMT203
003622  071236                          EMT1
003624  071362                          EDT1
003626  071452                          EFT1

401
402                             ;ERROR  204     "VV" WAS NOT RESET BY MAINTENANCE UNIT READY
403
003630  067362                          EMT204
003632  071236                          EMT1
003634  071362                          EDT1
003636  071452                          EFT1

404
405                             ;ERROR  205     SELECTED DEVICE HAS A PERSISTENT "SKI" ERROR
406
003640  067400                          EMT205
003642  071236                          EMT1
```

```
003644  071362                          EDT1
003646  071452                          EFT1

407
408                             ;ERROR  206     "LBC" DID NOT SET DURING DIAGNOSTIC MODE
409
003650  067410                          EMT206
003652  071236                          EHT1
003654  071362                          EDT1
003656  071452                          EFT1

410
411                             ;ERROR  207     UNEXPECTED LOSS OF "MOL" - MEDIUM IS OFF LINE
412
003660  067420                          EMT207
003662  071236                          EHT1
003664  071362                          EDT1
003666  071452                          EFT1

413
414                             ;ERROR  210     UNEXPECTED LOSS OF VOLUME VALID - "VV" = 0
415
003670  067432                          EMT210
003672  071236                          EHT1
003674  071362                          EDT1
003676  071452                          EFT1

416
417                             ;ERROR  211     UNEXPECTED MECHANICAL MOTION - "PIP" = 1
418
003700  067440                          EMT211
003702  071236                          EHT1
003704  071362                          EDT1
003706  071452                          EFT1

419
420                             ;ERROR  212     UNEXPECTED DEVICE FAULT - "DVC" = 1
421
003710  067454                          EMT212
003712  071236                          EHT1
003714  071362                          EDT1
003716  071452                          EFT1

422
423                             ;ERROR  213     UNEXPECTED SEEK INCOMPLETE ERROR - "SKI"   1
424
003720  067470                          EMT213
003722  071236                          EHT1
003724  071362                          EDT1
003726  071452                          EFT1

425
426                             ;ERROR  214     DRIVE EXECUTED A RECALIBRATE WITH ERROR SET
427
003730  067500                          EMT214
003732  071250                          EHT2
003734  071372                          EDT2
```

```
        003736  071462                          EFT2

    428
    429                             ;ERROR  215     DRIVE DID NOT DETECT "IVC" ERROR DURING RECALIBRATE
    430
        003740  067520                          EMT215
        003742  071250                          EHT2
        003744  071372                          EDT2
        003746  071462                          EFT2

    431
    432                             ;ERROR  216     INCORRECT "IVC" STATUS
    433
        003750  067532                          EMT216
        003752  071236                          EHT1
        003754  071362                          EDT1
        003756  071452                          EFT1

    434
    435                             ;ERROR  217     INCORRECT "IAE" STATUS
    436
        003760  067542                          EMT217
        003762  071236                          EHT1
        003764  071362                          EDT1
        003766  071452                          EFT1

    437
    438                             ;ERROR 220      INCORRECT "WLE" STATUS
    439
        003770  067552                          EMT220
        003772  071236                          EHT1
        003774  071362                          EDT1
        003776  071452                          EFT1

    440
    441                             ;ERROR  221     INCORRECT "OPI" STATUS
    442
        004000  067562                          EMT221
        004002  071236                          EHT1
        004004  071362                          EDT1
        004006  071452                          EFT1

    443
    444                             ;ERROR  222     RM DID NOT DETECT RMR ERROR
    445
        004010  067572                          EMT222
        004012  071236                          EHT1
        004014  071362                          EDT1
        004016  071452                          EFT1

    446
    447                             ;ERROR  223     RM DID NOT DETECT PARITY ERROR ON MASSBUS CONTROL BUS
    448
        004020  067602                          EMT223
        004022  071274                          EHT223
        004024  071406                          EDT223
        004026  071476                          EFT223
```

```
449
450                                     ;ERROR  224     UNUSED
451
    004030  000000                                0
    004032  000000                                0
    004034  000000                                0
    004036  000000                                0

452
453                                     ;ERROR  225     UNUSED
454
    004040  000000                                0
    004042  000000                                0
    004044  000000                                0
    004046  000000                                0

455
456                                     ;ERROR  226     UNUSED
457
    004050  000000                                0
    004052  000000                                0
    004054  000000                                0
    004056  000000                                0

458
459                                     ;ERROR  227     UNUSED
460
    004060  000000                                0
    004062  000000                                0
    004064  000000                                0
    004066  000000                                0

461
462                                     ;ERROR  230     UNUSED
463
    004070  000000                                0
    004072  000000                                0
    004074  000000                                0
    004076  000000                                0

464
465                                     ;ERROR  231     UNUSED
466
    004100  000000                                0
    004102  000000                                0
    004104  000000                                0
    004106  000000                                0

467
468                                     ;ERROR  232     UNUSED
469
    004110  000000                                0
    004112  000000                                0
    004114  000000                                0
    004116  000000                                0
```

```
470
471                                     ;ERROR  233     UNUSED
472
    004120  000000                              0
    004122  000000                              0
    004124  000000                              0
    004126  000000                              0

473
474                                     ;ERROR  234     UNUSED
475
    004130  000000                              0
    004132  000000                              0
    004134  000000                              0
    004136  000000                              0

476
477                                     ;ERROR  235     UNUSED
478
    004140  000000                              0
    004142  000000                              0
    004144  000000                              0
    004146  000000                              0

479
480                                     ;ERROR  236     UNUSED
481
    004150  000000                              0
    004152  000000                              0
    004154  000000                              0
    004156  000000                              0

482
483                                     ;ERROR  237     UNUSED
484
    004160  000000                              0
    004162  000000                              0
    004164  000000                              0
    004166  000000                              0

485
486                                     ;ERROR  240     UNUSED
487
    004170  000000                              0
    004172  000000                              0
    004174  000000                              0
    004176  000000                              0

488
489                                     ;ERROR  241     UNUSED
490
    004200  000000                              0
    004202  000000                              0
    004204  000000                              0
    004206  000000                              0

491
```

```
492                                     ;ERROR  242     UNUSED
493
    004210  000000                                0
    004212  000000                                0
    004214  000000                                0
    004216  000000                                0

494
495                                     ;ERROR  243     UNUSED
496
    004220  000000                                0
    004222  000000                                0
    004224  000000                                0
    004226  000000                                0

497
498                                     ;ERROR  244     UNUSED
499
    004230  000000                                0
    004232  000000                                0
    004234  000000                                0
    004236  000000                                0

500
501                                     ;ERROR  245     UNUSED
502
    004240  000000                                0
    004242  000000                                0
    004244  000000                                0
    004246  000000                                0

503
504                                     ;ERROR  246     "ATA" NOT RESET BY GO WHEN "ERR" = 0
505
    004250  067656                                EMT246
    004252  071236                                EHT1
    004254  071362                                EDT1
    004256  071452                                EFT1

506
507                                     ;ERROR  247     "ATA" NOT RESET BY WRITING RMAS
508
    004260  067666                                EMT247
    004262  071236                                EHT1
    004264  071362                                EDT1
    004266  071452                                EFT1

509
510                                     ;ERROR  250     "ATA" WAS RESET BY GO WHEN "ERR" = 1
511
    004270  067700                                EMT250
    004272  071236                                EHT1
    004274  071362                                EDT1
    004276  071452                                EFT1

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

```
514
    004300  067714                          EMT251
    004302  071250                          EHT2
    004304  071372                          EDT2
    004306  071462                          EFT2

515
516                             ;ERROR  252     PROGRAM INTERRUPT SHOULD NOT HAVE BEEN GENERATED
517
    004310  067722                          EMT252
    004312  071250                          EHT2
    004314  071372                          EDT2
    004316  071462                          EFT2

518
519                             ;ERROR  253     OFFSET MODE WAS NOT RESET BY WRITING RMDC
520
    004320  067730                          EMT253
    004322  071236                          EHT1
    004324  071362                          EDT1
    004326  071452                          EFT1

521
522                             ;ERROR  254     INCORRECT "ILF" STATUS
523
    004330  067746                          EMT254
    004332  071236                          EHT1
    004334  071362                          EDT1
    004336  071452                          EFT1

524
525                             ;ERROR  255     INCORRECT "ATA" STATUS
526
    004340  067756                          EMT255
    004342  071236                          EHT1
    004344  071362                          EDT1
    004346  071452                          EFT1

527
528                             ;ERROR  256     INCORRECT "ILR" STATUS
529
    004350  067766                          EMT256
    004352  071306                          EHT256
    004354  071406                          EDT223
    004356  071476                          EFT223

530
531                             ;ERROR  257     INVALID IAE STATUS DURING SEARCH COMMAND
532
    004360  067776                          EMT257
    004362  071236                          EHT1
    004364  071362                          EDT1
    004366  071452                          EFT1

533
534                             ;ERROR  260     "IVC" WAS NOT DETECTED DURING SEARCH COMMAND
535
```

```
004370  070010                          EMT260
004372  071236                          EHT1
004374  071362                          EDT1
004376  071452                          EFT1

536
537                             ;ERROR  261     DRIVE EXECUTED SEARCH WITH ERROR SET
538
004400  070022                          EMT261
004402  071236                          EHT1
004404  071362                          EDT1
004406  071452                          EFT1

539
540                             ;ERROR  262     "LBC" ERROR NOT SET DURING DIAGNOSTIC MODE
541
004410  070042                          EMT262
004412  071236                          EHT1
004414  071362                          EDT1
004416  071452                          EFT1

542
543                             ;ERROR  263     "SKI" ERROR DURING SEARCH COMMAND
544
004420  070052                          EMT263
004422  071236                          EHT1
004424  071362                          EDT1
004426  071452                          EFT1

545
546                             ;ERROR  264     "IVC" ERROR DURING SEARCH - LOST VOLUME VALID
547
004430  070062                          EMT264
004432  071236                          EHT1
004434  071362                          EDT1
004436  071452                          EFT1

548
549                             ;ERROR  265     ERRONEOUS IVC ERROR DURING SEARCH - VOLUME IS VALID
550
004440  070102                          EMT265
004442  071236                          EHT1
004444  071362                          EDT1
004446  071452                          EFT1

551
552                             ;ERROR  266     DEVICE FAULT (DVC) DURING SEARCH
553
004450  070122                          EMT266
004452  071236                          EHT1
004454  071362                          EDT1
004456  071452                          EFT1

554
555                             ;ERROR  267     SKI SHOULD HAVE BEEN SET BECAUSE CYLINDER
556                             ;               ADDRESS IS TOO LARGE
557
```

```
004460  070134                          EMT267
004462  071236                          EHT1
004464  071362                          EDT1
004466  071452                          EFT1

558
559                             ;ERROR  270     OPI ERROR DURING SEARCH BECAUSE MOL = 0
560
004470  070152                          EMT270
004472  071236                          EHT1
004474  071362                          EDT1
004476  071452                          EFT1

561
562                             ;ERROR  271     OPI ERROR DURING SEARCH BECAUSE ON CYLINDER
563                             ;               DIDN'T DROP
564
004500  070166                          EMT271
004502  071236                          EHT1
004504  071362                          EDT1
004506  071452                          EFT1

565
566                             ;ERROR  272     LOST MOL DURING SEARCH, OPI IS NOT SET
567
004510  070204                          EMT272
004512  071236                          EHT1
004514  071362                          EDT1
004516  071452                          EFT1

568
569                             ;ERROR  273     PIP STIL SET AFTER SEARCH
570
004520  070222                          EMT273
004522  071236                          EHT1
004524  071362                          EDT1
004526  071452                          EFT1

571
572                             ;ERROR  274     PARITY ERROR OCCURRED WHILE WRITING REMOTE
573                             ;               REGISTERS BUT MXF DID NOT SET
574
004530  070240                          EMT274
004532  071236                          EHT1
004534  071362                          EDT1
004536  071452                          EFT1

575
576                             ;ERROR  275     MXF ERROR - COMPOSITE ERROR OCCURRED BEFORE DATA
577                             ;               COMMAND STARTED
578
004540  070256                          EMT275
004542  071236                          EHT1
004544  071362                          EDT1
004546  071452                          EFT1

579
```

```
                                        ;ERROR  276     "OPI" ERROR DURING DATA TRANSFER BECAUSE "MOL" WAS
                                        ;               ZERO
004550  070270                                  EMT276
004552  071236                                  EHT1
004554  071362                                  EDT1
004556  071452                                  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
004560  070304                                  EMT277
004562  071236                                  EHT1
004564  071362                                  EDT1
004566  071452                                  EFT1

                                        ;ERROR  300     "IVC" ERROR DURING DATA TRANSFR BECAUSE VOLUME
                                        ;               WAS NOT VALID
004570  070322                                  EMT300
004572  071236                                  EHT1
004574  071362                                  EDT1
004576  071452                                  EFT1

                                        ;ERROR  301     ERRONEOUS "IVC" ERROR DURING DATA TRANSFER - VOLUME
                                        ;               IS VALID
004600  070342                                  EMT301
004602  071236                                  EHT1
004604  071362                                  EDT1
004606  071452                                  EFT1

                                        ;ERROR  302     "ILR" ERROR DURING DATA TRANSFER
004610  070364                                  EMT302
004612  071236                                  EHT1
004614  071362                                  EDT1
004616  071452                                  EFT1

                                        ;ERROR  303     "ILF" ERROR DURING DATA TRANSFER
004620  070376                                  EMT303
004622  071236                                  EHT1
004624  071362                                  EDT1
004626  071452                                  EFT1

                                        ;ERROR  304     "RMR" ERROR DURING DATA TRANSFER
004630  070410                                  EMT304
004632  071236                                  EHT1
```

```
004634  071362                          EDT1
004636  071452                          EFT1

605
606                             ;ERROR  305     INCORRECT "IAE" STATUS DURING DATA TRANSFER
607
004640  070422                          EMT305
004642  071236                          EHT1
004644  071362                          EDT1
004646  071452                          EFT1

608
609                             ;ERROR  306     "SKI" ERROR DURING DATA TRANSFER
610
004650  070434                          EMT306
004652  071236                          EHT1
004654  071362                          EDT1
004656  071452                          EFT1

611
612                             ;ERROR  307     DRIVE DID NOT DETECT SKI ERROR DUE TO CYLINDER
613
004660  070444                          EMT307
004662  071236                          EHT1
004664  071362                          EDT1
004666  071452                          EFT1

614
615                             ;ERROR  310     DEVICE FAULT DURING DATA TRANSFER
616
004670  070464                          EMT310
004672  071236                          EHT1
004674  071362                          EDT1
004676  071452                          EFT1

617
618                             ;ERROR  311     LOSS OF BIT CLOCK DURING DATA TRANSFER
619
004700  070476                          EMT311
004702  071236                          EHT1
004704  071362                          EDT1
004706  071452                          EFT1

620
621                             ;ERROR  312     LOSS OF SYSTEM CLOCK DURING DATA TRANSFER
622
004710  070510                          EMT312
004712  071236                          EHT1
004714  071362                          EDT1
004716  071452                          EFT1

623
624                             ;ERROR  313     UNSAFE ERROR DURING DATA TRANSFER (DVC = 0)
625
004720  070522                          EMT313
004722  071236                          EHT1
004724  071362                          EDT1
```

```
004726  071452                          EFT1

626
627                             ;ERROR  314     DRIVE TIMING ERROR DURING DATA TRANSFER
628
004730  070542                          EMT314
004732  071236                          EHT1
004734  071362                          EDT1
004736  071452                          EFT1

629
630                             ;ERROR  315     WRITE LOCK ERROR
631
004740  070554                          EMT315
004742  071236                          EHT1
004744  071362                          EDT1
004746  071452                          EFT1

632
633                             ;ERROR  316     ERRONEOUS WRITE LOCK ERROR
634
004750  070566                          EMT316
004752  071236                          EHT1
004754  071362                          EDT1
004756  071452                          EFT1

635
636                             ;ERROR  317     HEADER CRC ERROR DURING DATA TRANSFER
637
004760  070600                          EMT317
004762  071236                          EHT1
004764  071362                          EDT1
004766  071452                          EFT1

638
639                             ;ERROR  320     FORMAT ERROR DURING DATA TRANSFER
640
004770  070610                          EMT320
004772  071236                          EHT1
004774  071362                          EDT1
004776  071452                          EFT1

641
642                             ;ERROR  321     HEADER COMPARE ERROR DURING DATA TRANSFER
643
005000  070620                          EMT321
005002  071236                          EHT1
005004  071362                          EDT1
005006  071452                          EFT1

644
645                             ;ERROR  322     HEADER ERRORS SHOULD NOT BE SET
646
005010  070630                          EMT322
005012  071236                          EHT1
005014  071362                          EDT1
005016  071452                          EFT1
```

```
647
648                                     ;ERROR  323     DATA CHECK ERROR DURING DATA TRANSFER
649
    005020  070636                              EMT323
    005022  071236                              EHT1
    005024  071362                              EDT1
    005026  071452                              EFT1

650
651                                     ;ERROR  324     CORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
652
    005030  070646                              EMT324
    005032  071236                              EHT1
    005034  071362                              EDT1
    005036  071452                              EFT1

653
654                                     ;ERROR  325     UNCORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
655
    005040  070660                              EMT325
    005042  071236                              EHT1
    005044  071362                              EDT1
    005046  071452                              EFT1

656
657                                     ;ERROR  326     DATA PARITY ERROR DURING READ COMMMAND
658
    005050  070672                              EMT326
    005052  071236                              EHT1
    005054  071362                              EDT1
    005056  071452                              EFT1

659
660                                     ;ERROR  327     OFFSET MODE NOT RESET BY WRITE COMMAND
661
    005060  070710                              EMT327
    005062  071236                              EHT1
    005064  071362                              EDT1
    005066  071452                              EFT1

662
663                                     ;ERROR  330     DATA PARITY ERROR DURING WRITE COMMAND
664
    005070  070722                              EMT330
    005072  071236                              EHT1
    005074  071362                              EDT1
    005076  071452                              EFT1

665
666                                     ;ERROR  331     WRITE CLOCK FALURE DURING WRITE COMMAND
667
    005100  070732                              EMT331
    005102  071236                              EHT1
    005104  071362                              EDT1
    005106  071452                              EFT1
```

```
                                        ;ERROR  332     DATA LATE ERROR DURING DATA TRANSFER
005110  070744                                  EMT332
005112  071236                                  EHT1
005114  071362                                  EDT1
005116  071452                                  EFT1

                                        ;ERROR  333     PIP STIL SET AFTER DATA TRANSFER - SKI = 0
005120  070756                                  EMT333
005122  071236                                  EHT1
005124  071362                                  EDT1
005126  071452                                  EFT1

                                        ;ERROR  334     LOST MOL DURING DATA TRANSFER - OPI = 0
005130  070774                                  EMT334
005132  071236                                  EHT1
005134  071362                                  EDT1
005136  071452                                  EFT1

                                        ;ERROR  335     LOST VOLUME VALID DURING DATA TRANSFER - IVC  = 0
005140  071012                                  EMT335
005142  071236                                  EHT1
005144  071362                                  EDT1
005146  071452                                  EFT1

                                        ;ERROR  336     DATA READ DOES NOT COMPARE WITH DATA WRITTEN
005150  071032                                  EMT336
005152  071320                                  EHT336
005154  071416                                  EDT336
005156  071506                                  EFT336

                                        ;ERROR  337     WRITE CHECK ERROR NOT DETECTED
005160  071042                                  EMT337
005162  071332                                  EHT337
005164  071426                                  EDT337
005166  071516                                  EFT337

                                        ;ERROR  340     WRITE CHECK ERROR AT UNEXPECTED ADDRESS
005170  071052                                  EMT340
005172  071320                                  EHT336
005174  071416                                  EDT336
005176  071506                                  EFT336
```

```
690                                     ;ERROR  341     INCORRECT DATA DURING WRITE CHECK ERROR
691
     005200  071064                               EMT341
     005202  071320                               EHT336
     005204  071416                               EDT336
     005206  071506                               EFT336

692
693                                     ;ERROR  342     "IVC" ERROR NOT DETECTED DURING DATA TRANSFER
694
     005210  071072                               EMT342
     005212  071236                               EHT1
     005214  071362                               EDT1
     005216  071452                               EFT1

695
696                                     ;ERROR  343     "FER" NOT DETECTED DURING DATA TRANSFER
697
     005220  071104                               EMT343
     005222  071236                               EHT1
     005224  071362                               EDT1
     005226  071452                               EFT1

698
699                                     ;ERROR  344     "HCE" NOT DETECTED DURING DATA TRANSFER
700
     005230  071116                               EMT344
     005232  071344                               EHT344
     005234  071436                               EDT344
     005236  071526                               EFT344

701
702                                     ;ERROR  345     "BSE" NOT DETECTED DURING DATA TRANSFER
703
     005240  071130                               EMT345
     005242  071236                               EHT1
     005244  071362                               EDT1
     005246  071452                               EFT1

704
705                                     ;ERROR  346     HEADER ERROR WAS DETECTED W/ HCI SET
706
     005250  071140                               EMT346
     005252  071236                               EHT1
     005254  071362                               EDT1
     005256  071452                               EFT1

707
708                                     ;ERROR  347     DATA TRANSFER NOT ABORTED W/ COMP ERROR SET
709
     005260  071154                               EMT347
     005262  071236                               EHT1
     005264  071362                               EDT1
     005266  071452                               EFT1

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

```
712
     005270  071166                          EMT350
     005272  071236                          EHT1
     005274  071362                          EDT1
     005276  071452                          EFT1

713
714                                 ;ERROR  351     "ATA" DID NOT SET DURING SEARCH
715
     005300  071204                          EMT351
     005302  071236                          EHT1
     005304  071362                          EDT1
     005306  071452                          EFT1

716
717                                 ;ERROR  352     PROGRAM TIMEOUT WHILE TESTING RMLA
718
     005310  071214                          EMT352
     005312  000000                          0
     005314  000000                          0
     005316  000000                          0

719
720                                 ;ERROR  353     LOOK AHEAD TEST FAILS
721
     005320  071220                          EMT353
     005322  071356                          EHT353
     005324  071450                          EDT353
     005326  071536                          EFT353

722
723                                 ;ERROR  354     BSE SHOULD NOT BE SET
724
     005330  071230                          EMT354
     005332  071236                          EHT1
     005334  071362                          EDT1
     005336  071452                          EFT1

725
726                                 ;PUT ERROR TABLE HERE
727
```

```
                        .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
                                        
005340  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE THE TIME OUT OCCURED
005342  005740                          TST     -(R0)           ;ADJUST PC -2
005344  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
005346  104401  005354                  TYPE    ,65$            ;;TYPE ASCIZ STRING
005352  000417                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
005412                          64$:
005412  010046                          MOV     R0,-(SP)        ;SETUP FOR TYPING OUT PC
005414  104402                          TYPOC
005416  000240                          NOP                     ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
                                                                ;TO STOP ON UNEXPECTED TIMEOUT.
                                        
                                        .SBTTL  START OF PROGRAM
                                        
005420  000240                  START:  NOP
005422  005227  000000                  INC     #0              ;TTY LOOP, WAIT FOR INCREMENT
005426  001375                          BNE     .-4             ;OF WORD
005430  000005                          RESET                   ;RESET THE WORLD
                                        
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
005432  012706  001114                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
005436  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
005440  022706  001154                  CMP     #SWR,R6         ;;DONE?
005444  001374                          BNE     .-6             ;;LOOP BACK IF NO
005446  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
005452  012737  060222  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005460  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005466  012737  060626  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005474  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005502  012737  062366  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005510  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005516  012737  062474  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005524  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005532  013737  032374  032366          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
005540  005037  001206                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005544  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005550  112737  000001  001131          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005556  012737  005556  001122          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005564  012737  005564  001124          MOV     #.,$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.
005572  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005576  012737  005632  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
005604  012737  177570  001154          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005612  012737  177570  001156          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005620  022777  177777  173326          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005626  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005630  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
005632  012716  005640          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
005636  000002                          RTI
005640  012737  000176  001154  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
005646  012737  000174  001156          MOV     #DISPREG,DISPLAY
```

```
   005654  012637  000004            66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

   005660  005037  001230                    CLR     $PASS           ;;CLEAR PASS COUNT
   005664  132737  000200  001243            BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
   005672  001403                            BEQ     67$             ;;YES,USE NON-APT SWITCH
   005674  012737  001244  001154            MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
   005702                            67$:
                                     ;SETUP "TIMEOUT" TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
   005702  012737  005340  000004            MOV     #BADTMO,ERRVEC  ;SETUP FOR UNEXPECTED TIMEOUT
   005710  012737  000300  000006            MOV     #PR6,ERRVEC+2   ;LEVEL 6
   005716  012746  000300                    MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
   005722  012746  005730                    MOV     #68$,-(SP)      ;;PUT NEW PC ON  STACK
   005726  000002                            RTI                     ;;POP NEW PC AND PS
   005730                            68$:


                                     .SBTTL  TYPE PROGRAM NAME
                                     ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
   005730  005227  177777                    INC     #-1             ;;FIRST TIME?
   005734  001056                            BNE     69$             ;;BRANCH IF NO
   005736  022737  032530  000042            CMP     #$ENDAD,@#42    ;;ACT-11?
   005744  001452                            BEQ     69$             ;;BRANCH IF YES
   005746  104401  006014                    TYPE    ,70$            ;;TYPE ASCIZ STRING
                                     .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
   005752  005737  000042                    TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
   005756  001012                            BNE     71$             ;;BRANCH IF YES
   005760  123727  001242  000001            CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
   005766  001406                            BEQ     71$             ;;BRANCH IF YES
   005770  023727  001154  000176            CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
   005776  001005                            BNE     72$             ;;BRANCH IF NO
   006000  104407                            GTSWR                   ;;GET SOFT-SWR SETTINGS
   006002  000403                            BR      72$
   006004  112737  000001  001150    71$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
   006012                            72$:
   006012  000427                            BR      69$             ;;GET OVER THE ASCIZ
                                     ;;70$:  .ASCIZ  <CRLF>@CZRMNAO - RM05/3/2 FUNCTIONAL TEST, PART 2@<CRLF>
   006072                            69$:


                                     ;THE FOLLOWING FINDS OUT THE PROGRAM CONTROL MODE:
                                     ;PAPER TAPE (MANUAL), ACT11, XXDP CHAIN OR DUMP

   006072  005037  001330                    CLR     XXDP            ;CLEAR 'XXDP' LOAD DEVICE STORAGE
   006076  122737  000016  000041            CMPB    #16,@#41        ;LOADED FROM AN RM05/3/2 ?
   006104  001160                            BNE     3$              ;BRANCH IF NOT
   006106  013737  000040  001330            MOV     @#40,XXDP       ;GET DEVICE INDICATOR AND NUMBER
   006114  122737  000007  001330            CMPB    #7,XXDP         ;IS IT A VALID NUMBER ?
   006122  103002                            BHIS    1$              ;YES
   006124  105037  001330                    CLRB    XXDP            ;NO, DEFAULT TO DRIVE 0
   006130  005737  000042            1$:     TST     @#42            ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
   006134  001425                            BEQ     2$              ;BR IF NEITHER
   006136  104401  006144                    TYPE    ,74$            ;;TYPE ASCIZ STRING
   006142  000412                            BR      73$             ;;GET OVER THE ASCIZ
                                     ;;74$:  .ASCIZ  <CRLF>/NOT TESTING DRIVE /
   006170                            73$:
   006170  005046                            CLR     -(SP)           ;CLEAR WORD ON STACK
   006172  113716  001330                    MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
   006176  104403                            TYPOS                   ;TYPE THE ADDRESS
   006200     001                            .BYTE   1               ;ONLY 1 CHARACTER
```

```
     44 006201     000                            .BYTE   0               ;SUPRESS LEADING ZEROS
     45 006202  104401  001217                    TYPE    ,$CRLF          ;CR-LF
     46 006206  000517                            BR      3$              ;GET NUMBER OF DRIVES
     47
     48 006210  005227  177777          2$:       INC     #-1             ;FIRST TIME THRU HERE ?
     49 006214  001114                            BNE     3$              ;NO
     50 006216  104401  006224                    TYPE    ,76$            ;;TYPE ASCIZ STRING
        006222  000410                            BR      75$             ;;GET OVER THE ASCIZ
                                        ;;76$:    .ASCIZ  <CRLF>/TO TEST DRIVE /
        006244                          75$:
     51 006244  005046                            CLR     -(SP)           ;CLEAR WORD ON STACK
     52 006246  113716  001330                    MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
     53 006252  104403                            TYPOS                   ;TYPE DRIVE ADDRESS
     54 006254     001                            .BYTE   1               ;ONLY 1 CHARACTER
     55 006255     000                            .BYTE   0               ;SUPRESS LEADING ZEROS
     56 006256  104401  006264                    TYPE    ,78$            ;;TYPE ASCIZ STRING
        006262  000431                            BR      77$             ;;GET OVER THE ASCIZ
                                        ;;78$:    .ASCIZ  /, HALT PROGRAM, REMOVE RRDP PACK AND REPLACE IT/<CRLF>
        006346                          77$:
     57 006346  104401  006354                    TYPE    ,79$            ;;TYPE ASCIZ STRING
        006352  000435                            BR      3$              ;;GET OVER THE ASCIZ
                                        ;;79$:    .ASCIZ  /WITH A WORK PACK, CLEAR LOCATION 40 AND RESTART PROGRAM./<CRLF>
        006446                          3$:
     61                                 ;CHECK FOR AUTO MODE OR STANDLONE MODE
     62 006446  105737  001150                    TSTB    $AUTOB          ;RUNNING IN AUTO MODE ?
     63 006452  001515                            BEQ     STANDALONE      ;BR IF NO
     64 006454  012737  000377  001300            MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
     65
     66                                 ;PROGRAM IS RUNNING IN AUTO MODE - SEE IF SIZING IS ALLOWED
     67 006462                          XSIZ:
     68 006462  132737  000200  001243            BITB    #BIT7,$ENVM     ;SIZING ALLOWED ?
     69 006470  001075                            BNE     7$              ;NO
     70
     71 006472  005001                            CLR     R1              ;START FROM DRIVE 0
     72 006474  013700  001276                    MOV     $BASE,R0        ;LOAD THE BASE ADDRESS
     73
     74 006500  136137  064506  001300  1$:       BITB    ATNTBL(R1),$DEVM        ;IS DEVICE PRESENT IN MAP ?
     75 006506  001462                            BEQ     6$              ;BR IF NO
     76 006510  012737  064305  006652            MOV     #LODEV,5$       ;GET ADDRESS OF LOAD DEVICE MESSAGE
     77 006516  005737  001330                    TST     XXDP            ;LOADED FROM RM05/3/2 ?
     78 006522  001403                            BEQ     2$              ;NO
     79 006524  123701  001330                    CMPB    XXDP,R1         ;IS THIS THE DRIVE ?
     80 006530  001435                            BEQ     3$              ;YES, TRY NEXT DRIVE
     81
     82 006532  012760  000040  000010  2$:       MOV     #CLR,RMCS2(R0)  ;CLEAR MASS BUS
     83 006540  010160  000010                    MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
     84 006544  005760  000012                    TST     RMDS(R0)        ;TRY TO ACCESS AN RM DRIVE REGISTER
     85 006550  012737  064325  006652            MOV     #NOTPRS,5$      ;GET ADDRESS OF NOT PRESENT MESSAGE
     86 006556  032760  010000  000010            BIT     #NED,RMCS2(R0)  ;IS DRIVE PRESENT ?
     87 006564  001017                            BNE     3$              ;BR IF NO
     88 006566  012737  064342  006652            MOV     #NOTAVL,5$      ;GET ADDRESS OF AVAILABLE MESSAGE
     89 006574  032760  004000  000000            BIT     #DVA,RMCS1(R0)  ;IS DRIVE AVAILABLE ?
     90 006602  001410                            BEQ     3$              ;BR IF NO
     91 006604  012737  064361  006652            MOV     #OFFLIN,5$      ;GET ADDRESS OF OFF LINE MESSAGE
     92 006612  032760  010000  000012            BIT     #MOL,RMDS(R0)   ;IS MEDIUM ON LINE ?
     93 006620  001401                            BEQ     3$              ;BR IF NO
     94 006622  000414                            BR      6$
```

```
 95
 96 006624  146137  064506  001300  3$:     BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
 97 006632  104401  001217          4$:     TYPE    ,$CRLF                  ;CR-LF
 98 006636  104401  064277                  TYPE    ,MSGDRV                 ;TYPE 'DRIVE'
 99 006642  010146                          MOV     R1,-(SP)                ;;SAVE R1 FOR TYPEOUT
    006644  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
    006646     002                          .BYTE   2                       ;;TYPE 2 DIGIT(S)
    006647     000                          .BYTE   0                       ;;SUPPRESS LEADING ZEROS
100 006650  104401                          TYPE                            ;TYPE ERROR MESSAGE
101 006652  000000          5$:     .WORD   0                               ;ADDRESS OF MESSAGE GOES HERE
102
103 006654  005201          6$:     INC     R1                              ;INCREMENT THE DRIVE ADDRESS
104 006656  020127  000007          CMP     R1,#7                           ;ALL DRIVES ARE CHECKED ?
105 006662  003706                  BLE     1$                              ;BRANCH IF NOT
106
107 006664  104401  001217  7$:     TYPE    ,$CRLF                  ;CR-LF
108 006670  005004                  CLR     R4                      ;THESE TWO LOOPS ARE ADDED TO
109 006672  005304                  DEC     R4                      ;WAIT FOR TTY TO FINISH TYPING.
110 006674  001376                  BNE     .-2
111 006676  005304                  DEC     R4
112 006700  001376                  BNE     .-2
113
114 006702  000137  007612          JMP     CMNSTART                ;JUMP TO COMMON START
115
```

```
                                .SBTTL  STANDALONE INPUT ROUTINES
                        
006706                          STANDALONE:
006706  004737  061036                JSR     PC,$TKINT       ;INITIALIZE CONSOLE
                        
006712  005227  177777                INC     #-1             ;FIRST TIME THRU HERE ?
006716  001426                      BEQ     3$              ;YES !!
                        
                                ;SEE IF THE USER WANTS TO KEEP SAME DEVICES FOR TESTING
006720                          1$:
006720  104401  063644                TYPE    ,CNSL00         ;MAINTAIN PREVIOUS PARAMETERS ?
006724  104411                      RDCHR                   ;GET RESPONSE
006726  012637  001176                MOV     (SP)+,$TMP1     ;ECHO RESPONSE
006732  104401  001176                TYPE    ,$TMP1
006736  123727  001176  000131        CMPB    $TMP1,#'Y       ;YES RESPONSE ?
006744  001004                      BNE     2$              ;NO!!
006746  104401  001217                TYPE    ,$CRLF          ;CR-LF
006752  000137  007612                JMP     CMNSTART        ;KEEP PREVIOUS PARAMETERS
                        
006756  123727  001176  000116  2$:   CMPB    $TMP1,#'N       ;NO RESPONSE ?
006764  001427                      BEQ     5$              ;YES, GET NEW PARAMETERS
006766  104401  064241                TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT '
006772  000752                      BR      1$              ;TRY AGAIN
                        
                                ;SEE IF OPERATOR WANTS HELP TEXT
006774                          3$:
006774  104401  063234                TYPE    ,MSHELP         ;WANT HELP ?
007000  104411                      RDCHR                   ;GET RESPONSE
007002  012637  001176                MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
007006  104401  001176                TYPE    ,$TMP1
007012  123727  001176  000131        CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ?
007020  001407                      BEQ     4$              ;YES
007022  123727  001176  000116        CMPB    $TMP1,#'N       ;WAS IT A NO RESPONSE ?
007030  001405                      BEQ     5$              ;YES
007032  104401  064241                TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT'
007036  000756                      BR      3$              ;TRY AGAIN
007040  104401  101206          4$:   TYPE    ,HELP           ;YES - TYPE HELP TEXT
                        
                                ;SEE IF USER WANTS TO CHANGE ADDRESSES
007044                          5$:
007044  104401  001217                TYPE    ,$CRLF          ;CR-LF
007050  104401  063611                TYPE    ,UBUSQST        ;WANT TO CHANGE ADDRESS ?
007054  104411                      RDCHR                   ;GET RESPONSE
007056  012637  001176                MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
007062  104401  001176                TYPE    ,$TMP1
007066  123727  001176  000131        CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ?
007074  001407                      BEQ     6$              ;YES
007076  123727  001176  000116        CMPB    $TMP1,#'N       ;WAS IT A NO RESPONSE ?
007104  001525                      BEQ     12$             ;YES
007106  104401  064241                TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT '
007112  000756                      BR      5$+4            ;TRY AGAIN
                        
                                ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, VECTOR ADDRESS AND INTERRUPT PRIORITY
007114                          6$:
007114  104401  063677                TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
007120  013746  001276                MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
007124  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
007126  104401  063225                  TYPE    ,QUES           ;TYPE " ? "
007132  104413                          RDOCT                   ;GET NEW BUS ADDRESS
007134  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN ?
007140  001412                          BEQ     8$              ;YES-SKIP TO NEXT ENTRY
007142  022737  160000  001176          CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE ?
007150  101403                          BLOS    7$              ;YES
007152  104401  063715                  TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
007156  000760                          BR      6$+4            ;TRY AGAIN
007160  013737  001176  001276  7$:     MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS

007166  104401  063757          8$:     TYPE    ,CNSL03
007172  005046                          CLR     -(SP)
007174  113716  001272                  MOVB    $VECT1,(SP)     ;GET CURRENT VECTOR ADDRESS
007200  104403                          TYPOS
007202     003                          .BYTE   3               ;TYPE 3 DIGITS
007203     000                          .BYTE   0               ;SUPPRESS LEADING ZEROS
007204  104401  063225                  TYPE    ,QUES           ;TYPE " ? "
007210  104413                          RDOCT                   ;GET NEW VECTOR ADDRESS
007212  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN?
007216  001412                          BEQ     10$             ;YES-SKIP TO NEXT ENTRY
007220  022737  001000  001176          CMP     #1000,$TMP1     ;VECTOR ADDRESS < 1000 ?
007226  101003                          BHI     9$              ;YES!!
007230  104401  063777                  TYPE    ,CNSL04         ;TYPE WARNING MESSAGE
007234  000754                          BR      8$              ;RETRY
007236  113737  001176  001272  9$:     MOVB    $TMP1,$VECT1    ;STORE NEW VECTOR ADDRESS

007244  104401  064033          10$:    TYPE    ,CNSL05
007250  005046                          CLR     -(SP)
007252  113716  001273                  MOVB    $VECT1+1,(SP)   ;GET CURRENT BR LEVEL
007256  006216                          ASR     (SP)
007260  006216                          ASR     (SP)
007262  006216                          ASR     (SP)
007264  006216                          ASR     (SP)
007266  006216                          ASR     (SP)
007270  104403                          TYPOS
007272     001                          .BYTE   1               ;ONLY 1 DIGIT
007273     000                          .BYTE   0               ;SUPPRESS LEADING ZEROS
007274  104401  063225                  TYPE    ,QUES
007300  104413                          RDOCT                   ;GET NEW PRIORITY
007302  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN ?
007306  001424                          BEQ     12$             ;YES-SKIP TO NEXT ENTRY
007310  023727  001176  000007          CMP     $TMP1,#7        ;LEGAL PRIORITY ?
007316  002403                          BLT     11$             ;YES!!
007320  104401  064045                  TYPE    ,CNSL06         ;TYPE WARNING MESSAGE
007324  000747                          BR      10$             ;TRY AGAIN
007326  006337  001176          11$:    ASL     $TMP1           ;STORE NEW PRIORITY
007332  006337  001176                  ASL     $TMP1
007336  006337  001176                  ASL     $TMP1
007342  006337  001176                  ASL     $TMP1
007346  006337  001176                  ASL     $TMP1
007352  113737  001176  001273          MOVB    $TMP1,$VECT1+1


                                ;DIALOGUE TO INPUT DEVICE NUMBERS
007360                          12$:
007360  005227  177777                  INC     #-1             ;FIRST TIME THRU ?
007364  001002                          BNE     13$             ;BR IF NO
007366  104401  064076                  TYPE    ,CNSL07         ;TYPE INPUT INSTRUCTIONS
```

```
114 007372  104401  001217          13$:    TYPE    ,$CRLF          ;CR-LF
115 007376  005037  001300          14$:    CLR     $DEVM           ;CLEAR DEVICE MAP
116 007402  104401  064263                  TYPE    ,MSDRVS         ;TYPE 'DRIVE(S): '
117 007406  104411                          RDCHR
118 007410  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
119 007414  023727  001176  000101          CMP     $TMP1,#'A       ;IS INPUT "A" ?
120 007422  001007                          BNE     15$             ;NO
121 007424  104401  063220                  TYPE    ,ALL            ;YES, TYPE "ALL" AND GO
122 007430  012737  000377  001300          MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
123 007436  000137  006462                  JMP     XSIZ            ;AUTO SIZE.
124
125 007442  023727  001176  000015  15$:    CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
126 007450  001436                          BEQ     17$             ;YES
127 007452  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
128 007456  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
129 007464  002430                          BLT     17$             ;YES
130 007466  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
131 007474  003427                          BLE     18$             ;NO
132 007476  000423                          BR      17$             ;ILLEGAL INPUT
133
134 007500  104411                  16$:    RDCHR
135 007502  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
136 007506  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
137 007514  001432                          BEQ     19$             ;YES
138 007516  104401  063231                  TYPE    ,COMMA          ;TYPE ', '
139 007522  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
140 007526  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
141 007534  002404                          BLT     17$             ;YES
142 007536  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
143 007544  003403                          BLE     18$             ;NO
144 007546  104401  064241          17$:    TYPE    ,CNSL08         ;TYPE " ?ILLEGAL INPUT"
145 007552  000711                          BR      14$             ;RETRY
146
147 007554  013701  001176          18$:    MOV     $TMP1,R1        ;R1 = DRIVE NUMBER
148 007560  042701  177770                  BIC     #^C7,R1
149 007564  156137  064506  001300          BISB    ATNTBL(R1),$DEVM        ;SET DEVICE IN MAP
150 007572  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ?
151 007600  101337                          BHI     16$             ;NO
152 007602  104401  001217          19$:    TYPE    ,$CRLF          ;CR-LF
153 007606  000137  006462                  JMP     XSIZ            ;GO SIZE DEVICES
154
```

```
                                ;ASSEMBLE TEST QUE FROM DEVICE MAP
007612                          CMNSTART:
007612  013700  001300                  MOV     $DEVM,R0        ;R0 = DEVICE MAP
007616  012701  001466                  MOV     #TSTQUE+2,R1    ;R1 = ADDRESS OF FIRST ENTRY IN QUE
007622  010137  001464                  MOV     R1,TSTQUE       ;INITIALIZE ENTRY POINTER
007626  012702  000001                  MOV     #1,R2           ;R2 = DEVICE POINTER
007632  005003                          CLR     R3              ;R3 = DEVICE NUMBER
007634  030200                  1$:     BIT     R2,R0           ;IS THIS DEVICE IN MAP ?
007636  001406                          BEQ     2$              ;NO !!
007640  010311                          MOV     R3,(R1)         ;YES - ENTER DEVICE NUMBER IN QUE
007642  116361  064506  000001          MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
007650  062701  000002                  ADD     #2,R1           ;ADVANCE ENTRY POINTER
007654  006302                  2$:     ASL     R2              ;ADVANCE DEVICE POINTER
007656  105702                          TSTB    R2              ;DONE ALL DEVICES ?
007660  001402                          BEQ     3$              ;YES
007662  005203                          INC     R3              ;ADVANCE DEVICE NUMBER
007664  000763                          BR      1$              ;ENTER NEXT DEVICE
007666  005011                  3$:     CLR     (R1)            ;TERMINATE TEST QUE

                                ;SIZE FOR CLOCK
007670  004737  037600                  JSR     PC,SIZCLK       ;SEE IF CLOCK PRESENT
007674  000413                          BR      5$              ;YES - CLOCK IS PRESENT
007676                          4$:
007676  104000                          EMT
007700  104401  007706                  TYPE    ,65$            ;;TYPE ASCIZ STRING
007704  000405                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/PROG HLT/
007720                          64$:
007720  000000                          HALT                    ;PROGRAM HALT !!
007722  000765                          BR      4$
007724                          5$:
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
007724  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
007730  001012                          BNE     66$             ;;BRANCH IF YES
007732  123727  001242  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
007740  001406                          BEQ     66$             ;;BRANCH IF YES
007742  023727  001154  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
007750  001005                          BNE     67$             ;;BRANCH IF NO
007752  104407                          GTSWR                   ;;GET SOFT-SWR SETTINGS
007754  000403                          BR      67$
007756  112737  000001  001150  66$:     MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
007764                          67$:

007764  000240                  READY:  NOP                     ;READY TO START TEST
007766  105737  001300                  TSTB    $DEVM           ;ANY DRIVES IN MAP ?
007772  001007                          BNE     2$              ;BR IF YES
007774  005737  000042                  TST     @#42            ;ANY MONITOR PRESENT ?
010000  001002                          BNE     1$              ;BR IF YES
010002  000137  005420                  JMP     START           ;JUMP TO START
010006  000137  032340          1$:     JMP     $EOP            ;RETURN CONTROL TO MONITOR

010012  105037  001116          2$:     CLRB    $TSTNM          ;RESET TEST NUMBER
010016  005037  001206                  CLR     $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
010022  005037  001326                  CLR     CTLFG           ;CLEAR CONTROL-C FLAG
010026  004737  061036                  JSR     PC,$TKINT       ;INITIALIZE TTY
010032  012746  000300                  MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
010036  012746  010044                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
```

```
 010042  000002                          RTI                     ;;POP NEW PC AND PS
 010044                          64$:
 010044  117737  171414  001234          MOVB    @TSTQUE,$UNIT   ;LOAD UNIT NUMBER
 010052  005037  001510                  CLR     MEDENB          ;CLEAR MEDIA ENABLE

                                ;CLEAR MASSBUS CONTROLLER, SELECT DRIVE AND DETERMINE THE LAST TRACK
                                ;OF THE DIFFERENT DRIVE TYPES
 010056  012737  002000  001332          MOV     #TA4,LSTRK      ;ASSUME LAST TRACK FOR RM02/3 = 4.
 010064  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
 010070  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
 010076  117760  171362  000010          MOVB    @TSTQUE,RMCS2(R0)       ;SELECT DEVICE UNDER TEST
 010104  016002  000026                  MOV     RMDT(R0),R2     ;GET RMDT AND
 010110  042702  177770                  BIC     #177770,R2      ;SAVE DRIVE TYPE BITS
 010114  022702  000007                  CMP     #7,R2           ;IS IT AN RM05 ?
 010120  001003                          BNE     3$              ;NO, MUST BE AN RM02 OR RM03
 010122  012737  011000  001332          MOV     #TA16 TA2,LSTRK ;YES--SET LAST TRACK = '8.
 010130                          3$:
```

```
                                         ;;*****************************************************************
                                         ;*TEST 1          CONTROLLER ACCESS TEST
                                         ;;*****************************************************************
   010130                                TST1:
   010130  000004                                SCOPE                           ;SCOPE CALL
   010132  000240                                NOP                             ;START OF TEST
   010134  012706  001100                        MOV     #STACK,SP               ;INITIALIZE STACK POINTER
   010140  013700  001276                        MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
   010144  013701  001464                        MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
   010150  012737  000001  001226                MOV     #1,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX

   010156  005001                                CLR     R1
   010160  013746  000004                        MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
   010164  013746  000006                        MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
   010170  012737  010262  000004                MOV     #3$,ERRVEC
   010176  012737  000300  000006                MOV     #PR6,ERRVEC+2

   010204  110160  000001                        MOVB    R1,RMCS1+1(R0)  ;MOVE HI BYTE TO RMCS1
   010210  010160  000002                        MOV     R1,RMWC(R0)     ;MOVE WORD COUNT REGISTER
   010214  016002  000002                        MOV     RMWC(R0),R2
   010220  010160  000004                        MOV     R1,RMBA(R0)     ;MOVE BUS ADDRESS REGISTER
   010224  016002  000004                        MOV     RMBA(R0),R2
   010230  010160  000010                        MOV     R1,RMCS2(R0)    ;MOVE CONTROL STATUS REGISTER
   010234  016002  000010                        MOV     RMCS2(R0),R2
   010240  010160  000022                        MOV     R1,RMDB(R0)     ;MOVE DATA BUFFER
   010244  016002  000022                        MOV     RMDB(R0),R2
   010250  012637  000006                        MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
   010254  012637  000004                        MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
   010260  000415                                BR      7$              ;NO BUS TIMEOUT OCCURRED

   010262  022626                        3$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
   010264  012637  000006                        MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
   010270  012637  000004                        MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
   010274  104110                                EMT     110
   010276  005737  000042                        TST     @#42            ;STAND ALONE MODE ?
   010302  001002                                BNE     5$              ;NO'.
   010304  000137  005420                        JMP     START           ;YES-GO GET $BASE

   010310  000137  032340                5$:     JMP     $EOP            ;GO TO END OF PASS HANDLER
   010314                                7$:


                                         ;;*****************************************************************
                                         ;*TEST 2          DEVICE AVAILABLE TEST
                                         ;;*****************************************************************
   010314                                TST2:
   010314  000004                                SCOPE                           ;SCOPE CALL
   010316  000240                                NOP                             ;START OF TEST
   010320  012706  001100                        MOV     #STACK,SP               ;INITIALIZE STACK POINTER
   010324  013700  001276                        MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
   010330  013701  001464                        MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
   010334  012737  000002  001226                MOV     #2,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX

   010342  004737  046464                        JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
   010346  000404                                BR      2$              ;GO TO 2$ IF NO ERROR
   010350  000240                                NOP                     ;RETURN HERE IF ERROR
   010352  104000                                EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
```

```
010354  000137  010474                   JMP     7$                 ;GO TO 7$ IF ERROR
010360                           2$:
010360  013746  000004                   MOV     ERRVEC,-(SP)       ;;PUSH ERRVEC ON STACK
010364  013746  000006                   MOV     ERRVEC+2,-(SP)             ;;PUSH ERRVEC+2 ON STACK
010370  012737  010460  000004           MOV     #5$,ERRVEC
010376  013737  000300  000^C6           MOV     PR6,ERRVEC+2

010404  016037  000000  001176           MOV     RMCS1(R0),$TMP1    ;GET DVA STATUS
010412  016037  000010  001174           MOV     RMCS2(R0),$TMP0    ;GET NED STATUS
010420  012637  000006                   MOV     (SP)+,ERRVEC+2             ;;POP STACK INTO ERRVEC+2
010424  012637  000004                   MOV     (SP)+,ERRVEC       ;;POP STACK INTO ERRVEC
010430  032737  010000  001174           BIT     #NED,$TMP0         ;NONEXISTENT DEVICE ?
010436  001402                           BEQ     3$                 ;NO!!
010440  104111                           EMT     111
010442  000414                           BR      7$
010444  032737  004000  001176   3$:     BIT     #DVA,$TMP1         ;DEVICE AVAILABLE ?
010452  001012                           BNE     9$                 ;YES
010454  104112                           EMT     112
010456  000406                           BR      7$

010460  022626                   5$:     CMP     (SP)+,(SP)+        ;ADJUST STACK
010462  012637  000006                   MOV     (SP)+,ERRV:C+2             ;;POP STACK INTO ERRVEC+2
010466  012637  000004                   MOV     (SP)+,ERRVEC       ;;POP STACK INTO ERRVEC
010472  104113                           EMT     113
010474  000137  032302           7$:     JMP     $EOSP
010500                           9$:


                                 ;;*****************************************************************
                                 ;*TEST 3          DRIVE TVPE TEST
                                 ;;*****************************************************************
010500                           TST3:
010500  000004                           SCOPE                      ;SCOPE CALL
010502  000240                           NOP                        ;START OF TEST
010504  012706  001100                   MOV     #STACK,SP          ;INITIALIZE STACK POINTER
010510  013700  001276                   MOV     $BASE,R0           ;R0 = UNIBUS ADDRESS
010514  013701  001464                   MOV     TSTQUE,R1          ;(R1) = DEVICE BEING TESTED
010520  012737  000003  001226           MOV     #3,$TES^N          ;;SET TEST NUMBER IN APT MAIL BOX

010526  004737  046464                   JSR     PC,CNTCLR          ;GO ISSUE CONTROLLER CLEAR
010532  000404                           BR      2$                 ;GO TO 2$ IF NO ERROR
010534  000240                           NOP                        ;RETURN HERE IF ERROR
010536  104000                           EMT                        ;ERROR NUMBER DEFINED BY SUBROUTINE
010540  000137  010676                   JMP     4$                 ;GO TO 4$ IF ERROR
010544                           2$:
010544  112737  000026  001522           MOVB    #RMDT,GETINX       ;SETUP GET INDEX TABLE
010552  112737  000200  001523           MOVB    #200,GETINX+1      ;SETUP TERMINATOR BYTE
010560  012737  010702  001362           MOV     #5$,RMDTI          ;SET RMDT INPUT BUFFER = 5$
010566  004737  037110                   JSR     PC,GET             ;GO READ RMDT VIA GET SUBROUTINE
010572  000402                           BR      3$                 ;GO TO 3$ IF NO ERROR
010574  000240                           NOP                        ;RETURN HERE IF ERROR
010576  104000                           EMT                        ;ERROR DEFINED BY GET SUBROUTINE

010600  022737  020024  001362   3$:     CMP     #SNGPRT,RMDTI      ;SINGLE PORT RM03 ?
010606  001435                           BEQ     5$                 ;YES !!
010610  022737  024024  001362           CMP     #DULPRT,RMDTI      ;DUAL PORT RM03 ?
010616  001431                           BEQ     5$                 ;YES !!
```

```
 74 010620  022737  020025  001362          CMP     #SNGPRT BIT0,RMDTI      ;SINGLE PORT RM02 ?
 75 010626  001425                          BEQ     5$                      ;YES !!
 76 010630  022737  024025  001362          CMP     #DULPRT BIT0,RMDTI      ;DUAL PORT RM02 ?
 77 010636  001421                          BEQ     5$                      ;YES !!
 78
 79 010640  022737  020027  001362          CMP     #SNGPRT.BIT1 BIT0,RMDTI ;SINGLE PORT RM05 ?
 80 010646  001415                          BEQ     5$                      ;YES !!
 81 010650  022737  024027  001362          CMP     #DULPRT!BIT1!BIT0,RMDTI ;DUAL PORT RM05 ?
 82 010656  001411                          BEQ     5$
 83
 84 010660  012737  020024  001176          MOV     #SNGPRT,$TMP1   ;LOAD ACCEPTABLE VALUES
 85 010666  012737  024024  001200          MOV     #DULPRT,$TMP2
 86 010674  104114                          EMT     114
 87
 88 010676  000137  032302          4$:     JMP     $EOSP           ;GO TO SUBPASS HANDLER.
 89 010702                          5$:
 90
 91                                 ;;*****************************************************************
                                    ;*TEST 4         FORMAT ZEROS
                                    ;;*****************************************************************
    010702                          TST4:
    010702  000004                          SCOPE                   ;SCOPE CALL
    010704  000240                          NOP                     ;START OF TEST
    010706  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
    010712  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    010716  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
    010722  012737  000004  001226          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 92
 93                                 ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
 94 010730  012737  000000  001442          MOV     #0,RMOFO        ;18 BIT FORMAT
 95 010736                          5$:
 96 010736  012737  000000  001444          MOV     #0.,RMDCO       ;CYLINDER = 0
 97 010744  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
 98 010752  012737  177376  001412          MOV     #-258.,RMWCO    ;2 * 256 WORDS (2'S COMP)
 99 010760  012737  101206  001414          MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
100 010766  012737  000062  001410          MOV     #WH,RMCS1O      ;WRITE HEADER AND DATA
101
102                                 ;VERIFY THAT SECTOR IS NOT BAD
    010774  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    011000  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
    011002  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    011006  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    011010  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
103 011014                          10$:
104 011014  012737  064620  001174          MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
105 011022  012737  000001  001176          MOV     #1,$TMP1
106 011030  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
107 011034                          20$:
108
109                                 ;PREPARE DEVICE FOR DATA TRANSFER
110 011034  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    011040  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
     011042  000404                    BR      30$             ;GO TO 30$ IF NO ERROR
     011044  000240                    NOP                     ;RETURN HERE IF ERROR
     011046  104000                    EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     011050  000137  011544            JMP     190$            ;GO TO 190$ IF ERROR
111  011054                    30$:
112
113                            ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
114  011054  012737  000005  001410    MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
115  011062  012702  001551            MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
116  011066  112722  000006            MOVB    #RMDA,(R2)+
117  011072  112722  000034            MOVB    #RMDC,(R2)+
118  011076  112722  000032            MOVB    #RMOF,(R2)+
119  011102  112722  000000            MOVB    #RMCS1,(R2)+
120  011106  112722  000200            MOVB    #200,(R2)+
121
122  011112  004737  037360            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     011116  000404                    BR      40$             ;GO TO 40$ IF NO ERROR
     011120  000240                    NOP                     ;RETURN HERE IF ERROR
     011122  104000                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     011124  000137  011544            JMP     190$            ;GO TO 190$ IF ERROR
123  011130                    40$:
124
125                            ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
126  011130  004737  037024            JSR     PC,GETSTS       ;SETUP FOR STATUS
127  011134  004737  037722            JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
128  011140                    50$:
129
130                            ;GO READ SEEK STATUS
131  011140  004737  037110            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     011144  000404                    BR      60$             ;GO TO 60$ IF NO ERROR
     011146  000240                    NOP                     ;RETURN HERE IF ERROR
     011150  104000                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     011152  000137  011544            JMP     190$            ;GO TO 190$ IF ERROR
132  011156                    60$:
133
134                            ;VERIFY THE RESULTS OF THE SEEK COMMAND
135  011156  004737  045224            JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     011162  000405                    BR      70$             ;GO TO 70$ IF NO ERROR
     011164  000240                    NOP                     ;RETURN HERE IF ERROR
     011166  104000                    EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     011170  004736                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     011172  000137  011544            JMP     190$            ;GO TO 190$ IF ERROR
136  011176                    70$:
137
138                            ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
139  011176  012737  000063  001410    MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
140  011204  012702  001554            MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
141  011210  112722  000002            MOVB    #RMWC,(R2)+
142  011214  112722  000004            MOVB    #RMBA,(R2)+
143  011220  112722  000000            MOVB    #RMCS1,(R2)+
144  011224  112722  000200            MOVB    #200,(R2)+
145
146  011230  004737  037360            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     011234  000404                    BR      80$             ;GO TO 80$ IF NO ERROR
     011236  000240                    NOP                     ;RETURN HERE IF ERROR
```

```
        011240  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        011242  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    147 011246                          80$:
    148
    149                                 ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
    150 011246  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
    151
    152 011252  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        011256  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        011260  000240                          NOP                     ;RETURN HERE IF ERROR
        011262  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        011264  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    153 011270                          90$:
    154
    155                                 ;VERIFY RESULTS OF WRITE COMMAND
    156 011270  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        011274  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
        011276  000240                          NOP                     ;RETURN HERE IF ERROR
        011300  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        011302  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011304  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    157 011310                          100$:
    158 011310  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        011314  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        011316  000240                          NOP                     ;RETURN HERE IF ERROR
        011320  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        011322  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011324  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    159 011330                          110$:
    160 011330  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        011334  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        011336  000240                          NOP                     ;RETURN HERE IF ERROR
        011340  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        011342  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011344  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    161 011350                          120$:
    162
    163                                 ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
    164 011350  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
    165 011356  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
    166
    167 011364  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        011370  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        011372  000240                          NOP                     ;RETURN HERE IF ERROR
        011374  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        011376  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    168 011402                          130$:
    169
    170                                 ;WAIT FOR READ TO COMPLETE AND GET STATUS
    171 011402  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
    172
    173 011406  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        011412  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        011414  000240                          NOP                     ;RETURN HERE IF ERROR
        011416  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        011420  000137  011544                  JMP     190$            ;GO TO 190$ IF ERROR
    174 011424                          140$:
```

```
                                        ;VERIFY THE RESULTS OF READ OPERATION
011424  004737  040106                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
011430  000405                                BR      150$            ;GO TO 150$ IF NO ERROR
011432  000240                                NOP                     ;RETURN HERE IF ERROR
011434  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
011436  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011440  000137  011544                        JMP     190$            ;GO TO 190$ IF ERROR
011444                                  150$:
011444  004737  052622                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
011450  000405                                BR      160$            ;GO TO 160$ IF NO ERROR
011452  000240                                NOP                     ;RETURN HERE IF ERROR
011454  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
011456  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011460  000137  011544                        JMP     190$            ;GO TO 190$ IF ERROR
011464                                  160$:
011464  004737  040740                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
011470  000405                                BR      170$            ;GO TO 170$ IF NO ERROR
011472  000240                                NOP                     ;RETURN HERE IF ERROR
011474  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
011476  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011500  000137  011504                        JMP     190$            ;GO TO 190$ IF ERROR
011504                                  170$:

                                        ;VERIFY DATA
011504  004737  036462                        JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
011510  101206                                .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
011512  102212                                .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
011514  000402                                BR      180$            ;GO TO 180$ IF NO ERROR
011516  000240                                NOP                     ;RETURN HERE IF ERROR
011520  104000                                EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
011522                                  180$:
011522  032737  010000  001442                BIT     #FMT16,RMOFO    ;TEST 16 BIT MODE YET ?
011530  001005                                BNE     190$            ;YES
011532  012737  010000  001442                MOV     #FMT16,RMOFO    ;SET 16 BIT MODE AND
011540  000137  010736                        JMP     5$              ;TEST AGAIN.
011544                                  190$:

                                        ;;*****************************************************************
                                        ;*TEST 5        ZERO FILL TEST
                                        ;;*****************************************************************
011544                                  TST5:
011544  000004                                SCOPE                   ;SCOPE CALL
011546  000240                                NOP                     ;START OF TEST
011550  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
011554  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
011560  013701  001464                        MOV     TSTQUE.R1       ;(R1) = DEVICE BEING TESTED
011564  012737  000005  001226                MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

                                        ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
011572  012737  000000  001444                MOV     #0,RMDCO        ;CYLINDER = 0
011600  012737  000000  001416                MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
011606  012737  010000  001442                MOV     #FMT16,RMOFO    ;16 BIT FORMAT
011614  012737  177376  001412                MOV     #-258.,RMWCO    ;2 + 256 WORDS (2'S COMP)
011622  012737  101206  001414                MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
011630  012737  000062  001410                MOV     #WH,RMCS1O      ;WRITE HEADER AND DATA
```

```
                                        ;VERIFY THAT SECTOR IS NOT BAD
011636  004737  034276                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
011642  000405                                BR      10$             ;GO TO 10$ IF NO ERROR
011644  104401  063120                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
011650  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
011652  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
011656                                  10$:
011656  012737  064620  001174                MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
011664  012737  000001  001176                MOV     #1,$TMP1
011672  004737  036224                        JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
011676                                  20$:

                                        ;PREPARE DEVICE FOR DATA TRANSFER
011676  004737  033352                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
011702  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
011704  000404                                BR      30$             ;GO TO 30$ IF NO ERROR
011706  000240                                NOP                     ;RETURN HERE IF ERROR
011710  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
011712  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
011716                                  30$:

                                        ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
011716  012737  000005  001410                MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
011724  012702  001551                        MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
011730  112722  000006                        MOVB    #RMDA,(R2)+
011734  112722  000034                        MOVB    #RMDC,(R2)+
011740  112722  000032                        MOVB    #RMOF,(R2)+
011744  112722  000000                        MOVB    #RMCS1,(R2)+
011750  112722  000200                        MOVB    #200,(R2)+

011754  004737  037360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011760  000404                                BR      40$             ;GO TO 40$ IF NO ERROR
011762  000240                                NOP                     ;RETURN HERE IF ERROR
011764  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
011766  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
011772                                  40$:

                                        ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
011772  004737  037024                        JSR     PC,GETSTS       ;SETUP FOR STATUS
011776  004737  037722                        JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
012002                                  50$:

                                        ;GO READ SEEK STATUS
012002  004737  037110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012006  000404                                BR      60$             ;GO TO 60$ IF NO ERROR
012010  000240                                NOP                     ;RETURN HERE IF ERROR
012012  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
012014  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012020                                  60$:
```

```
                                        ;VERIFY THE RESULTS OF THE SEEK COMMAND
012020  004737  045224                        JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
012024  000405                                BR      70$             ;GO TO 70$ IF NO ERROR
012026  000240                                NOP                     ;RETURN HERE IF ERROR
012030  104000                                EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
012032  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012034  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012040                                  70$:

                                        ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
012040  012737  177776  001412              MOV     #-2,RMWCO       ;FORMAT PARTIAL SECTOR
012046  012737  000063  001410              MOV     #WH!GO,RMCS1O   ;WRITE HEADER AND DATA
012054  012702  001554                      MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
012060  112722  000002                      MOVB    #RMWC,(R2)+
012064  112722  0000C.                      MOVB    #RMBA,(R2)+
012070  112722  000000                      MOVB    #RMCS1,(R2)+
012074  112722  000200                      MOVB    #200,(R2)+

012100  004737  037360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
012104  000404                              BR      80$             ;GO TO 80$ IF NO ERROR
012106  000240                              NOP                     ;RETURN HERE IF ERROR
012110  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
012112  000137  012400                      JMP     180$            ;GO TO 180$ IF ERROR
012116                                  80$:

                                        ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
012116  004737  037722                      JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

012122  004737  037110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012126  000404                              BR      90$             ;GO TO 90$ IF NO ERROR
012130  000240                              NOP                     ;RETURN HERE IF ERROR
012132  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
012134  000137  012400                      JMP     180$            ;GO TO 180$ IF ERROR
012140                                  90$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
012140  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
012144  000405                              BR      100$            ;GO TO 100$ IF NO ERROR
012146  000240                              NOP                     ;RETURN HERE IF ERROR
012150  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
012152  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012154  000137  012400                      JMP     180$            ;GO TO 180$ IF ERROR
012160                                  100$:
012160  004737  052622                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
012164  000405                              BR      110$            ;GO TO 110$ IF NO ERROR
012166  000240                              NOP                     ;RETURN HERE IF ERROR
012170  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
012172  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012174  000137  012400                      JMP     180$            ;GO TO 180$ IF ERROR
012200                                  110$:
012200  004737  040740                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
012204  000405                              BR      120$            ;GO TO 120$ IF NO ERROR
012206  000240                              NOP                     ;RETURN HERE IF ERROR
012210  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
012212  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012214  000137  012400                      JMP     180$            ;GO TO 180$ IF ERROR
012220                                  120$:
```

```
                                        ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
012220  012737  177376  001412                MOV     #-258.,RMWCO    ;2 + 256 WORDS (2'S COMP)
012226  012737  000073  001410                MOV     #RH!GO,RMCS1O   ;READ HEADER & DATA COMMAND
012234  012737  102212  001414                MOV     #BUFTWO,RMBAO   ;CHANGE BUS ADDRESS

012242  004737  037360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
012246  000404                                BR      130$            ;GO TO 130$ IF NO ERROR
012250  000240                                NOP                     ;RETURN HERE IF ERROR
012252  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
012254  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012260                                130$:

                                        ;WAIT FOR READ TO COMPLETE AND GET STATUS
012260  004737  037722                        JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE

012264  004737  037110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012270  000404                                BR      140$            ;GO TO 140$ IF NO ERROR
012272  000240                                NOP                     ;RETURN HERE IF ERROR
012274  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
012276  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012302                                140$:

                                        ;VERIFY THE RESULTS OF READ OPERATION
012302  004737  040106                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
012306  000405                                BR      150$            ;GO TO 150$ IF NO ERROR
012310  000240                                NOP                     ;RETURN HERE IF ERROR
012312  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
012314  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012316  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012322                                150$:
012322  004737  052622                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
012326  000405                                BR      160$            ;GO TO 160$ IF NO ERROR
012330  000240                                NOP                     ;RETURN HERE IF ERROR
012332  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
012334  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012336  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012342                                160$:
012342  004737  040740                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
012346  000405                                BR      170$            ;GO TO 170$ IF NO ERROR
012350  000240                                NOP                     ;RETURN HERE IF ERROR
012352  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
012354  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012356  000137  012400                        JMP     180$            ;GO TO 180$ IF ERROR
012362                                170$:

                                        ;VERIFY DATA
012362  004737  036462                        JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
012366  101206                                .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
012370  102212                                .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
012372  000402                                BR      180$            ;GO TO 180$ IF NO ERROR
012374  000240                                NOP                     ;RETURN HERE IF ERROR
012376  104000                                EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
012400                                180$:

                                        ;;*******************************************************************
                                        ;*TEST 6         FORMAT CHECK ZEROS
```

```
                                        ;;*****************************************************************
012400                                  TST6:
012400  000004                                SCOPE                   ;SCOPE CALL
012402  000240                                NOP                     ;START OF TEST
012404  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
012410  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
012414  013701  001464                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
012420  012737  000006  001226                MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
292
293                                     ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
294 012426  012737  000000  001444            MOV     #0,RMDC0        ;CYLINDER = 0
295 012434  012737  000000  001416            MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
296 012442  012737  010000  001442            MOV     #FMT16,RMOF0    ;16 BIT FORMAT
297 012450  012737  177376  001412            MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
298 012456  012737  101206  001414            MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
299 012464  012737  000062  001410            MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
300
301                                     ;VERIFY THAT SECTOR IS NOT BAD
012472  004737  034276                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
012476  000405                                BR      10$             ;GO TO 10$ IF NO ERROR
012500  104401  063120                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
012504  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
012506  000137  013170                        JMP     260$            ;GO TO 260$ IF ERROR
302 012512                              10$:
303 012512  012737  064620  001174            MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
304 012520  012737  000001  001176            MOV     #1,$TMP1
305 012526  004737  036224                    JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
306 012532                              20$:
307
308                                     ;PREPARE DEVICE FOR DATA TRANSFER
309 012532  004737  033352                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
012536  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
012540  000404                                BR      30$             ;GO TO 30$ IF NO ERROR
012542  000240                                NOP                     ;RETURN HERE IF ERROR
012544  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
012546  000137  013170                        JMP     260$            ;GO TO 260$ IF ERROR
310 012552                              30$:
311
312                                     ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
313 012552  012737  000005  001410            MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
314 012560  012702  001551                    MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
315 012564  112722  000006                    MOVB    #RMDA,(R2)+
316 012570  112722  000034                    MOVB    #RMDC,(R2)+
317 012574  112722  000032                    MOVB    #RMOF,(R2)+
318 012600  112722  000000                    MOVB    #RMCS1,(R2)+
319 012604  112722  000200                    MOVB    #200,(R2)+
320
321 012610  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
012614  000404                                BR      40$             ;GO TO 40$ IF NO ERROR
012616  000240                                NOP                     ;RETURN HERE IF ERROR
```

```
        012620  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        012622  000137  013170                  JMP     260$            ;GO TO 260$ IF ERROR
    322 012626                          40$:
    323
    324                                 ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
    325 012626  004737  037024                  JSR     PC,GETSTS       ;SETUP FOR STATUS
    326 012632  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
    327 012636                          50$:
    328
    329                                 ;GO READ SEEK STATUS
    330 012636  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        012642  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
        012644  000240                          NOP                     ;RETURN HERE IF ERROR
        012646  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        012650  000137  013170                  JMP     260$            ;GO TO 260$ IF ERROR
    331 012654                          60$:
    332
    333                                 ;VERIFY THE RESULTS OF THE SEEK COMMAND
    334 012654  004737  045224                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        012660  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        012662  000240                          NOP                     ;RETURN HERE IF ERROR
        012664  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        012666  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012670  000137  013170                  JMP     260$            ;GO TO 260$ IF ERROR
    335 012674                          70$:
    336
    337                                 ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
    338 012674  012737  000063  001410          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
    339 012702  012702  001554                  MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
    340 012706  112722  000002                  MOVB    #RMWC,(R2)+
    341 012712  112722  000004                  MOVB    #RMBA,(R2)+
    342 012716  112722  000000                  MOVB    #RMCS1,(R2)+
    343 012722  112722  000200                  MOVB    #200,(R2)+
    344
    345 012726  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        012732  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        012734  000240                          NOP                     ;RETURN HERE IF ERROR
        012736  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        012740  000137  013170                  JMP     260$            ;GO TO 260$ IF ERROR
    346 012744                          80$:
    347
    348                                 ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
    349 012744  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
    350
    351 012750  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        012754  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        012756  000240                          NOP                     ;RETURN HERE IF ERROR
        012760  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        012762  000137  013170                  JMP     260$            ;GO TO 260$ IF ERROR
    352 012766                          90$:
    353
    354                                 ;VERIFY RESULTS OF WRITE COMMAND
    355 012766  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        012772  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
        012774  000240                          NOP                     ;RETURN HERE IF ERROR
        012776  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        013000  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
013002  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
356 013006                      100$:
357 013006  004737  052622              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
013012  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
013014  000240                          NOP                     ;RETURN HERE IF ERROR
013016  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
013020  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013022  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
358 013026                      110$:
359 013026  004737  040740              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
013032  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
013034  000240                          NOP                     ;RETURN HERE IF ERROR
013036  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
013040  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013042  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
360 013046                      120$:
361
362                             ;WRITE CHECK HEADER AND DATA FOR SECTOR JUST WRITTEN
363 013046  012737  000053  001410      MOV     #WCH!GO,RMCS10  ;WRITE CHECK COMMAND
364
365 013054  004737  037360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
013060  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
013062  000240                          NOP                     ;RETURN HERE IF ERROR
013064  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
013066  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
366 013072                      130$:
367
368                             ;WAIT FOR WRITE CHECK TO COMPLETE AND GET STATUS
369 013072  004737  037722              JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
370
371 013076  004737  037110              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
013102  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
013104  000240                          NOP                     ;RETURN HERE IF ERROR
013106  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
013110  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
372 013114                      140$:
373
374                             ;VERIFY THE RESULTS OF WRITE CHECK OPERATION
375 013114  004737  040106              JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
013120  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
013122  000240                          NOP                     ;RETURN HERE IF ERROR
013124  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
013126  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013130  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
376 013134                      150$:
377 013134  004737  052622              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
013140  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
013142  000240                          NOP                     ;RETURN HERE IF ERROR
013144  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
013146  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013150  000137  013170                 JMP     260$            ;GO TO 260$ IF ERROR
378 013154                      160$:
379 013154  004737  040740              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
013160  000403                          BR      170$            ;GO TO 170$ IF NO ERROR
013162  000240                          NOP                     ;RETURN HERE IF ERROR
013164  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
013166  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
                                        170$:
                                        260$:
                                        ;;*****************************************************************
                                        ;*TEST 7        FORMAT CHECK ZEROS W/ WCE ERROR
                                        ;;*****************************************************************
013170                                  TST7:
013170  000004                                  SCOPE                   ;SCOPE CALL
013172  000240                                  NOP                     ;START OF TEST
013174  012706  001100                          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
013200  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
013204  013701  001464                          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
013210  012737  000007  001226                  MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

                                        ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
013216  012737  000000  001444                  MOV     #0,RMDC0        ;CYLINDER = 0
013224  012737  000000  001416                  MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
013232  012737  010000  001442                  MOV     #FMT16,RMOF0    ;16 BIT FORMAT
013240  012737  177376  001412                  MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
013246  012737  101206  001414                  MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
013254  012737  000062  001410                  MOV     #WH,RMCS10      ;WRITE HEADER AND DATA

                                        ;VERIFY THAT SECTOR IS NOT BAD
013262  004737  034276                          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
013266  000405                                  BR      10$             ;GO TO 10$ IF NO ERROR
013270  104401  063120                          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
013274  104000                                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
013276  000137  014130                          JMP     260$            ;GO TO 260$ IF ERROR
013302                                  10$:
013302  012737  064620  001174                  MOV     #ZEROS,$TMP0    ;USE ALL ONES DATA PATTERN
013310  012737  000001  001176                  MOV     #1,$TMP1
013316  004737  036224                          JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
013322                                  20$:

                                        ;PREPARE DEVICE FOR DATA TRANSFER
013322  004737  033352                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
013326  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
013330  000404                                  BR      30$             ;GO TO 30$ IF NO ERROR
013332  000240                                  NOP                     ;RETURN HERE IF ERROR
013334  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
013336  000137  014130                          JMP     260$            ;GO TO 260$ IF ERROR
013342                                  30$:

                                        ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
013342  012737  000005  001410                  MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
013350  012702  001551                          MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
013354  112722  000006                          MOVB    #RMDA,(R2)+
013360  112722  000034                          MOVB    #RMDC,(R2)+
013364  112722  000032                          MOVB    #RMOF,(R2)+
```

```
                                                    MOVB    #RMCS1,(R2)+
                                                    MOVB    #200,(R2)+

                                                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                                    BR      40$             ;GO TO 40$ IF NO ERROR
                                                    NOP                     ;RETURN HERE IF ERROR
                                                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                                    JMP     260$            ;GO TO 260$ IF ERROR
                                                40$:

                                                ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
                                                    JSR     PC,GETSTS       ;SETUP FOR STATUS
                                                    JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
                                                50$:

                                                ;GO READ SEEK STATUS
                                                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                    BR      60$             ;GO TO 60$ IF NO ERROR
                                                    NOP                     ;RETURN HERE IF ERROR
                                                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                                    JMP     260$            ;GO TO 260$ IF ERROR
                                                60$:

                                                ;VERIFY THE RESULTS OF THE SEEK COMMAND
                                                    JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
                                                    BR      70$             ;GO TO 70$ IF NO ERROR
                                                    NOP                     ;RETURN HERE IF ERROR
                                                    EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
                                                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                                    JMP     260$            ;GO TO 260$ IF ERROR
                                                70$:

                                                ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
                                                    MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
                                                    MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
                                                    MOVB    #RMWC,(R2)+
                                                    MOVB    #RMBA,(R2)+
                                                    MOVB    #RMCS1,(R2)+
                                                    MOVB    #200,(R2)+

                                                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                                    BR      80$             ;GO TO 80$ IF NO ERROR
                                                    NOP                     ;RETURN HERE IF ERROR
                                                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                                    JMP     260$            ;GO TO 260$ IF ERROR
                                                80$:

                                                ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
                                                    JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

                                                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                    BR      90$             ;GO TO 90$ IF NO ERROR
                                                    NOP                     ;RETURN HERE IF ERROR
                                                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                                    JMP     260$            ;GO TO 260$ IF ERROR
                                                90$:
```

```
                                        ;VERIFY RESULTS OF WRITE COMMAND
013556  004737  040106                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
013562  000405                                BR      100$            ;GO TO 100$ IF NO ERROR
013564  000240                                NOP                     ;RETURN HERE IF ERROR
013566  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
013570  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013572  000137  014130                        JMP     260$            ;GO TO 260$ IF ERROR
013576                                  100$:
013576  004737  052622                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
013602  000405                                BR      110$            ;GO TO 110$ IF NO ERROR
013604  000240                                NOP                     ;RETURN HERE IF ERROR
013606  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
013610  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013612  000137  014130                        JMP     260$            ;GO TO 260$ IF ERROR
013616                                  110$:
013616  004737  040740                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
013622  000405                                BR      120$            ;GO TO 120$ IF NO ERROR
013624  000240                                NOP                     ;RETURN HERE IF ERROR
013626  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
013630  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013632  000137  014130                        JMP     260$            ;GO TO 260$ IF ERROR
013636                                  120$:

                                        ;ALTER DATA BUFFER
013636  005137  102210                        COM     BUFTWO-2        ;COMPLEMENT LAST DATA WORD

                                        ;SETUP AND WRITE CHECK HEADER AND DATA COMMAND
013642  012737  000053  001410                MOV     #WCH!GO,RMCS10  ;WRITE CHECK COMMAND

013650  004737  037360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
013654  000404                                BR      180$            ;GO TO 180$ IF NO ERROR
013656  000240                                NOP                     ;RETURN HERE IF ERROR
013660  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
013662  000137  014130                        JMP     260$            ;GO TO 260$ IF ERROR
013666                                  180$:

                                        ;WAIT FOR WRITE CHECK COMMAND TO COMPLETE AND READ STATUS
013666  004737  037722                        JSR     PC,TIMOUT       ;WAIT FOR COMMMAND TO COMPLETE

013672  004737  037110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
013676  000404                                BR      190$            ;GO TO 190$ IF NO ERROR
013700  000240                                NOP                     ;RETURN HERE IF ERROR
013702  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
013704  000137  014130                        JMP     260$            ;GO TO 260$ IF ERROR
013710                                  190$:

                                        ;CHECK FOR PRIMARY ERRORS
013710  004737  040106                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
013714  000405                                BR      200$            ;GO TO 200$ IF NO ERROR
013716  000240                                NOP                     ;RETURN HERE IF ERROR
013720  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
013722  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013724  000137  014130                        JMP     260$            ;GO TO 260$ IF ERROR
013730                                  200$:

                                        ;MAKE SURE THE WRITE CHECK ERROR WAS DETECTED
013730  032737  040000  001344                BIT     #WCE,RMCS21     ;IS WRITE CHECK ERROR SET??
```

```
                                        BNE     210$            ;YES!!
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      205$            ;GO TO 205$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     260$            ;GO TO 260$ IF ERROR

001140  205$:   MOV     RMCS2I,$GDDAT   ;LOAD EXPECTED STATUS
001140          BIS     #WCE,$GDDAT
001142          MOV     RMCS2I,$BDDAT   ;LOAD RECEIVED STATUS
                EMT     337
                BR      260$
        210$:

                ;VERIFY THE ADDRESS OF THE WRITE CHECK ERROR
001134          MOV     #BUFTWO-2,$GDADR;LOAD EXPECTED ADDRESS
001136          MOV     RMBAI,$BDADR    ;LOAD RECEIVED ADDRESS
001136          SUB     #2,$BDADR       ;DECREMENT RECEIVED ADDRESS

                ;GET WCE DATA AND VERIFY IT IS OK
001522          MOVB    #RMDB,GETINX    ;SETUP FOR READING RMDB
001523          MOVB    #200,GETINX+1

                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                BR      230$            ;GO TO 230$ IF NO ERROR
                NOP                     ;RETURN HERE IF ERROR
                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                JMP     260$            ;GO TO 260$ IF ERROR
001142  230$:   MOV     RMDBI,$BDDAT    ;LOAD RECEIVED DATA WORD
001140          MOV     BUFTWO-2,$GDDAT ;LOAD EXPECTED DATA WORD
001140          COM     $GDDAT
001136          CMP     $GDADR,$BDADR   ;IS ADDRESS OK??
                BEQ     220$            ;YES!!
                EMT     340
                BR      260$
001142  220$:   CMP     $GDDAT,$BDDAT   ;IS DATA WORD OK??
                BEQ     260$            ;YES!!
                EMT     341
        260$:

                ;;*******************************************************************
                ;*TEST 10        FORMAT ONES
                ;;*******************************************************************
        TST10:
                SCOPE                   ;SCOPE CALL
                NOP                     ;START OF TEST
                MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
001226          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
001444          MOV     #0,RMDC0        ;CYLINDER = 0
001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
```

```
014200  012737  177376  001412          MOV     #-258.,RMWCO    ;2 + 256 WORDS (2'S COMP)
014206  012737  101206  001414          MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
014214  012737  000062  001410          MOV     #WH,RMCS10      ;WRITE HEADER AND DATA

                                ;VERIFY THAT SECTOR IS NOT BAD
014222  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
014226  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
014230  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
014234  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
014236  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
014242                          10$:
014242  012737  064556  001174          MOV     #ONES,$TMPO     ;USE ALL ONES DATA PATTERN
014250  012737  000001  00'176          MOV     #1,$TMP1
014256  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
014262                          20$:

                                ;PREPARE DEVICE FOR DATA TRANSFER
014262  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
014266  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
014270  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
014272  000240                          NOP                     ;RETURN HERE IF ERROR
014274  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
014276  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
014302                          30$:

                                ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
014302  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
014310  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
014314  112722  000006                  MOVB    #RMDA,(R2)+
014320  112722  000034                  MOVB    #RMDC,(R2)+
014324  112722  000032                  MOVB    #RMOF,(R2)+
014330  112722  000000                  MOVB    #RMCS1,(R2)+
014334  112722  000200                  MOVB    #200,(R2)+

014340  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
014344  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
014346  000240                          NOP                     ;RETURN HERE IF ERROR
014350  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
014352  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
014356                          40$:

                                ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
014356  004737  037024                  JSR     PC,GETSTS       ;SETUP FOR STATUS
014362  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
014366                          50$:

                                ;GO READ SEEK STATUS
014366  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
014372  000404                          BF      60$             ;GO TO 60$ IF NO ERROR
014374  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        014376  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014400  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
    549 014404                          60$:
    550
    551                                 ;VERIFY THE RESULTS OF THE SEEK COMMAND
    552 014404  004737  045224                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        014410  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        014412  000240                          NOP                     ;RETURN HERE IF ERROR
        014414  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        014416  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014420  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
    553 014424                          70$:
    554
    555                                 ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
    556 014424  012737  000063  001410          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
    557 014432  012702  001554                  MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
    558 014436  112722  000002                  MOVB    #RMWC,(R2)+
    559 014442  112722  000004                  MOVB    #RMBA,(R2)+
    560 014446  112722  000000                  MOVB    #RMCS1,(R2)+
    561 014452  112722  000200                  MOVB    #200,(R2)+
    562
    563 014456  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        014462  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        014464  000240                          NOP                     ;RETURN HERE IF ERROR
        014466  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        014470  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
    564 014474                          80$:
    565
    566                                 ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
    567 014474  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
    568
    569 014500  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        014504  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        014506  000240                          NOP                     ;RETURN HERE IF ERROR
        014510  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014512  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
    570 014516                          90$:
    571
    572                                 ;VERIFY RESULTS OF WRITE COMMAND
    573 014516  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        014522  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
        014524  000240                          NOP                     ;RETURN HERE IF ERROR
        014526  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        014530  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014532  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
    574 014536                          100$:
    575 014536  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        014542  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        014544  000240                          NOP                     ;RETURN HERE IF ERROR
        014546  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        014550  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014552  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
    576 014556                          110$:
    577 014556  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        014562  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        014564  000240                          NOP                     ;RETURN HERE IF ERROR
        014566  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
```

```
                014570  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                014572  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
578             014576                          120$:
579
580                                             ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
581             014576  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
582             014604  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
583
584             014612  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                014616  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
                014620  000240                          NOP                     ;RETURN HERE IF ERROR
                014622  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                014624  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
585             014630                          130$:
586
587                                             ;WAIT FOR READ TO COMPLETE AND GET STATUS
588             014630  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
589
590             014634  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                014640  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
                014642  000240                          NOP                     ;RETURN HERE IF ERROR
                014644  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                014646  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
591             014652                          140$:
592
593                                             ;VERIFY THE RESULTS OF READ OPERATION
594             014652  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                014656  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                014660  000240                          NOP                     ;RETURN HERE IF ERROR
                014662  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                014664  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                014666  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
595             014672                          150$:
596             014672  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                014676  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
                014700  000240                          NOP                     ;RETURN HERE IF ERROR
                014702  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                014704  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                014706  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
597             014712                          160$:
598             014712  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                014716  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
                014720  000240                          NOP                     ;RETURN HERE IF ERROR
                014722  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                014724  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                014726  000137  014750                  JMP     180$            ;GO TO 180$ IF ERROR
599             014732                          170$:
600
601                                             ;VERIFY DATA
602             014732  004737  036462                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
                014736  101206                          .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
                014740  102212                          .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
                014742  000402                          BR      180$            ;GO TO 180$ IF NO ERROR
                014744  000240                          NOP                     ;RETURN HERE IF ERROR
                014746  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
603             014750                          180$:
604
```

```
605                                     ;;*****************************************************************
                                        ;*TEST 11       FORMAT CHECK ONES
                                        ;;*****************************************************************
        014750                          TST11:
        014750  000004                          SCOPE                   ;SCOPE CALL
        014752  000240                          NOP                     ;START OF TEST
        014754  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        014760  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        014764  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        014770  012737  000011  001226          MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
606
607                                     ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
608 014776  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
609 015004  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
610 015012  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
611 015020  012737  177376  001412          MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
612 015026  012737  101206  001414          MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
613 015034  012737  000062  001410          MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
614
615                                     ;VERIFY THAT SECTOR IS NOT BAD
        015042  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        015046  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
        015050  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        015054  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        015056  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
616 015062                              10$:
617 015062  012737  064556  001174          MOV     #ONES,$TMP0     ;USE ALL ONES DATA PATTERN
618 015070  012737  000001  001176          MOV     #1,$TMP1
619 015076  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
620 015102                              20$:
621
622                                     ;PREPARE DEVICE FOR DATA TRANSFER
623 015102  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        015106  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
        015110  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        015112  000240                          NOP                     ;RETURN HERE IF ERROR
        015114  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        015116  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
624 015122                              30$:
625
626                                     ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
627 015122  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
628 015130  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
629 015134  112722  000006                  MOVB    #RMDA,(R2)+
630 015140  112722  000034                  MOVB    #RMDC,(R2)+
631 015144  112722  000032                  MOVB    #RMOF,(R2)+
632 015150  112722  000000                  MOVB    #RMCS1,(R2)+
633 015154  112722  000200                  MOVB    #200,(R2)+
634
635 015160  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
```

```
        015164  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        015166  000240                          NOP                     ;RETURN HERE IF ERROR
        015170  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        015172  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    636 015176                          40$:
    637
    638
    639 015176  004737  037024                  JSR     PC,GETSTS       ;SETUP FOR STATUS
    640 015202  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
    641 015206                          50$:
    642
    643
    644 015206  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015212  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
        015214  000240                          NOP                     ;RETURN HERE IF ERROR
        015216  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015220  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    645 015224                          60$:
    646
    647
    648 015224  004737  045224                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        015230  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        015232  000240                          NOP                     ;RETURN HERE IF ERROR
        015234  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        015236  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015240  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    649 015244                          70$:
    650
    651
    652 015244  012737  000063  001410          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
    653 015252  012702  001554                  MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
    654 015256  112722  000002                  MOVB    #RMWC,(R2)+
    655 015262  112722  000004                  MOVB    #RMBA,(R2)+
    656 015266  112722  000000                  MOVB    #RMCS1,(R2)+
    657 015272  112722  000200                  MOVB    #200,(R2)+
    658
    659 015276  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        015302  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        015304  000240                          NOP                     ;RETURN HERE IF ERROR
        015306  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        015310  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    660 015314                          80$:
    661
    662
    663 015314  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
    664
    665 015320  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015324  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        015326  000240                          NOP                     ;RETURN HERE IF ERROR
        015330  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015332  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    666 015336                          90$:
    667
    668
    669 015336  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        015342  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
        015344  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        015346  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        015350  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015352  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    670 015356                          100$:
    671 015356  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        015362  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        015364  000240                          NOP                     ;RETURN HERE IF ERROR
        015366  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        015370  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015372  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    672 015376                          110$:
    673 015376  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        015402  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        015404  000240                          NOP                     ;RETURN HERE IF ERROR
        015406  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        015410  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015412  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    674 015416                          120$:
    675
    676                                 ;WRITE CHECK HEADER AND DATA FOR SECTOR JUST WRITTEN
    677 015416  012737  000053  001410          MOV     #WCH!GO,RMCS10  ;WRITE CHECK COMMAND
    678
    679 015424  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        015430  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        015432  000240                          NOP                     ;RETURN HERE IF ERROR
        015434  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        015436  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    680 015442                          130$:
    681
    682                                 ;WAIT FOR WRITE CHECK TO COMPLETE AND GET STATUS
    683 015442  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
    684
    685 015446  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015452  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        015454  000240                          NOP                     ;RETURN HERE IF ERROR
        015456  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015460  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    686 015464                          140$:
    687
    688                                 ;VERIFY THE RESULTS OF WRITE CHECK OPERATION
    689 015464  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        015470  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        015472  000240                          NOP                     ;RETURN HERE IF ERROR
        015474  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        015476  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015500  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    690 015504                          150$:
    691 015504  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        015510  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        015512  000240                          NOP                     ;RETURN HERE IF ERROR
        015514  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        015516  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015520  000137  015540                  JMP     260$            ;GO TO 260$ IF ERROR
    692 015524                          160$:
    693 015524  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        015530  000403                          BR      170$            ;GO TO 170$ IF NO ERROR
        015532  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
                015534  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                015536  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
694             015540                          170$:
695
696             015540                          260$:
697
698                                             ;;*****************************************************************
                                                ;*TEST 12        FORMAT CHECK ONES W/ WCE ERRORS
                                                ;;*****************************************************************
                015540                          TST12:
                015540  000004                          SCOPE                   ;SCOPE CALL
                015542  000240                          NOP                     ;START OF TEST
                015544  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                015550  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                015554  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                015560  012737  000012  001226          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
699
700                                             ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
701             015566  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
702             015574  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
703             015602  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
704             015610  012737  177376  001412          MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
705             015616  012737  101206  001414          MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
706             015624  012737  000062  001410          MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
707
708                                             ;VERIFY THAT SECTOR IS NOT BAD
                015632  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
                015636  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
                015640  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
                015644  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
                015646  000137  016476                  JMP     260$            ;GO TO 260$ IF ERROR
709             015652                          10$:
710             015652  012737  064556  001174          MOV     #ONES,$TMP0     ;USE ALL ONES DATA PATTERN
711             015660  012737  000001  001176          MOV     #1,$TMP1
712             015666  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
713             015672                          20$:
714
715                                             ;PREPARE DEVICE FOR DATA TRANSFER
716             015672  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                015676  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
                015700  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
                015702  000240                          NOP                     ;RETURN HERE IF ERROR
                015704  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                015706  000137  016476                  JMP     260$            ;GO TO 260$ IF ERROR
717             015712                          30$:
718
719                                             ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
720             015712  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
721             015720  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
722             015724  112722  000006                  MOVB    #RMDA,(R2)+
```

```
                                ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
                                ;GO READ SEEK STATUS
                                ;VERIFY THE RESULTS OF THE SEEK COMMAND
                                ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
                                ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
```

```
 723 015730  112722  000034                 MOVB    #RMDC,(R2)+
 724 015734  112722  000032                 MOVB    #RMOF,(R2)+
 725 015740  112722  000000                 MOVB    #RMCS1,(R2)+
 726 015744  112722  000200                 MOVB    #200,(R2)+
 727
 728 015750  004737  037360                 JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     015754  000404                         BR      40$             ;GO TO 40$ IF NO ERROR
     015756  000240                         NOP                     ;RETURN HERE IF ERROR
     015760  104000                         EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     015762  000137  016476                 JMP     260$            ;GO TO 260$ IF ERROR
 729 015766                          40$:
 730
 731
 732 015766  004737  037024                 JSR     PC,GETSTS       ;SETUP FOR STATUS
 733 015772  004737  037722                 JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
 734 015776                          50$:
 735
 736
 737 015776  004737  037110                 JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     016002  000404                         BR      60$             ;GO TO 60$ IF NO ERROR
     016004  000240                         NOP                     ;RETURN HERE IF ERROR
     016006  104000                         EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     016010  000137  016476                 JMP     260$            ;GO TO 260$ IF ERROR
 738 016014                          60$:
 739
 740
 741 016014  004737  045224                 JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     016020  000405                         BR      70$             ;GO TO 70$ IF NO ERROR
     016022  000240                         NOP                     ;RETURN HERE IF ERROR
     016024  104000                         EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     016026  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     016030  000137  016476                 JMP     260$            ;GO TO 260$ IF ERROR
 742 016034                          70$:
 743
 744
 745 016034  012737  000063  001410         MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
 746 016042  012702  001554                 MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
 747 016046  112722  000002                 MOVB    #RMWC,(R2)+
 748 016052  112722  000004                 MOVB    #RMBA,(R2)+
 749 016056  112722  000000                 MOVB    #RMCS1,(R2)+
 750 016062  112722  000200                 MOVB    #200,(R2)+
 751
 752 016066  004737  037360                 JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     016072  000404                         BR      80$             ;GO TO 80$ IF NO ERROR
     016074  000240                         NOP                     ;RETURN HERE IF ERROR
     016076  104000                         EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     016100  000137  016476                 JMP     260$            ;GO TO 260$ IF ERROR
 753 016104                          80$:
 754
 755
 756 016104  004737  037722                 JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
 757
 758 016110  004737  037110                 JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     016114  000404                         BR      90$             ;GO TO 90$ IF NO ERROR
     016116  000240                         NOP                     ;RETURN HERE IF ERROR
     016120  104000                         EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     016122  000137  016476                 JMP     260$            ;GO TO 260$ IF ERROR
```

```
                                90$:

                                ;VERIFY RESULTS OF WRITE COMMAND
016126  004737  040106                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
016132  000405                        BR      100$            ;GO TO 100$ IF NO ERROR
016134  000240                        NOP                     ;RETURN HERE IF ERROR
016136  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
016140  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016142  000137  016476                JMP     260$            ;GO TO 260$ IF ERROR
016146                          100$:
016146  004737  052622                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
016152  000405                        BR      110$            ;GO TO 110$ IF NO ERROR
016154  000240                        NOP                     ;RETURN HERE IF ERROR
016156  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
016160  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016162  000137  016476                JMP     260$            ;GO TO 260$ IF ERROR
016166                          110$:
016166  004737  040740                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
016172  000405                        BR      120$            ;GO TO 120$ IF NO ERROR
016174  000240                        NOP                     ;RETURN HERE IF ERROR
016176  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
016200  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016202  000137  016476                JMP     260$            ;GO TO 260$ IF ERROR
016206                          120$:

                                ;ALTER DATA BUFFER
016206  005137  102210                COM     BUFTWO-2        ;COMPLEMENT DATA WORD

                                ;SETUP AND WRITE CHECK HEADER AND DATA COMMAND
016212  012737  000053  001410        MOV     #WCH.GO,RMCS10  ;WRITE CHECK COMMAND

016220  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
016224  000404                        BR      180$            ;GO TO 180$ IF NO ERROR
016226  000240                        NOP                     ;RETURN HERE IF ERROR
016230  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
016232  000137  016476                JMP     260$            ;GO TO 260$ IF ERROR
016236                          180$:

                                ;WAIT FOR WRITE CHECK COMMAND TO COMPLETE AND READ STATUS
016236  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR COMMMAND TO COMPLETE

016242  004737  037110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016246  000404                        BR      190$            ;GO TO 190$ IF NO ERROR
016250  000240                        NOP                     ;RETURN HERE IF ERROR
016252  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
016254  000137  016476                JMP     260$            ;GO TO 260$ IF ERROR
016260                          190$:

                                ;CHECK FOR PRIMARY ERRORS
016260  004737  040106                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
016264  000405                        BR      200$            ;GO TO 200$ IF NO ERROR
016266  000240                        NOP                     ;RETURN HERE IF ERROR
016270  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
016272  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016274  000137  016476                JMP     260$            ;GO TO 260$ IF ERROR
016300                          200$:
'87
```

```
                                        ;MAKE SURE THE WRITE CHECK ERROR WAS DETECTED
016300  032737  040000  001344            BIT     #WCE,RMCS2I     ;IS WRITE CHECK ERROR SET??
016306  001022                            BNE     210$            ;YES.!

016310  004737  052622                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
016314  000405                            BR      205$            ;GO TO 205$ IF NO ERROR
016316  000240                            NOP                     ;RETURN HERE IF ERROR
016320  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
016322  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016324  000137  016476                    JMP     260$            ;GO TO 260$ IF ERROR
016330  013737  001344  001140  205$:     MOV     RMCS2I,$GDDAT   ;LOAD EXPECTED STATUS
016336  052737  040000  001140            BIS     #WCE,$GDDAT
016344  013737  001344  001142            MOV     RMCS2I,$BDDAT   ;LOAD RECEIVED STATUS
016352  104337                            EMT     337
016354                            210$:

                                        ;VERIFY THE ADDRESS OF THE WRITE CHECK ERROR
016354  012737  102210  001134            MOV     #BUFTWO-2,$GDADR        ;LOAD EXPECTED ADDRESS
016362  013737  001340  001136            MOV     RMBAI,$BDADR    ;LOAD RECEIVED ADDRESS
016370  162737  000002  001136            SUB     #2,$BDADR       ;DECREMENT RECFIVED ADDRESS

                                        ;GET WCE DATA AND VERIFY IT IS OK
016376  112737  000022  001522            MOVB    #RMDB,GETINX    ;SETUP FOR READING RMDB
016404  112737  000200  001523            MOVB    #200,GETINX+1

016412  004737  037110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016416  000404                            BR      230$            ;GO TO 230$ IF NO ERROR
016420  000240                            NOP                     ;RETURN HERE IF ERROR
016422  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
016424  000137  016476                    JMP     260$            ;GO TO 260$ IF ERROR
016430  013737  001356  001142  230$:     MOV     RMDBI,$BDDAT    ;LOAD RECEIVED DATA WORD
016436  013737  102210  001140            MOV     BUFTWO-2,$GDDAT ;LOAD EXPECTED DATA WORD
016444  005137  001140                    COM     $GDDAT
016450  023737  001134  001136            CMP     $GDADR,$BDADR   ;IS ADDRESS OK??
016456  001402                            BEQ     220$            ;YES!!
016460  104340                            EMT     340
016462  000405                            BR      260$
016464                            220$:
016464  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;IS DATA WORD OK??
016472  001401                            BEQ     260$            ;YES!!
016474  104341                            EMT     341
016476                            260$:

                                        ;;*****************************************************************
                                        ;*TEST 13       FORMAT MULTIPLE SECTORS
                                        ;;*****************************************************************
016476                            TST13:
016476  000004                            SCOPE                   ;SCOPE CALL
016500  000240                            NOP                     ;START OF TEST
016502  012706  001100                    MOV     #STACK,SP       ;INITIALIZE STACK POINTER
016506  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
016512  013701  001464                    MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
016516  012737  000013  001226            MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                        ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
016524  012737  010000  001442            MOV     #FMT16,RMOFO    ;16 BIT FORMAT
016532  012737  000000  001444            MOV     #0,RMDCO        ;CYLINDER = 0
```

```
827 016540  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
828 016546  012737  176774  001412          MOV     #-258.*2,RMWCO  ;WORD COUNT FOR 2 SECTORS (2'S COMP)
829 016554  012737  101206  001414          MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
830 016562  012737  000062  001410          MOV     #WH,RMCS1O      ;WRITE HEADER AND DATA
831
832                                 ;VERIFY THAT SECTOR IS NOT BAD
    016570  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    016574  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
    016576  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    016602  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    016604  000137  017272                  JMP     170$            ;GO TO 170$ IF ERROR
833 016610                          10$:
834 016610  012737  064620  001174          MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
835 016616  012737  000001  001176          MOV     #1,$TMP1
836 016624  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
837 016630                          20$:
838
839                                 ;PREPARE DEVICE FOR DATA TRANSFER
840 016630  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    016634  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
    016636  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
    016640  000240                          NOP                     ;RETURN HERE IF ERROR
    016642  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    016644  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
841 016650                          30$:
842
843                                 ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
844 016650  012737  000005  001410          MOV     #SEEK!GO,RMCS1O ;CHANGE COMMAND TO SEEK
845 016656  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
846 016662  112722  000006                  MOVB    #RMDA,(R2)+
847 016666  112722  000034                  MOVB    #RMDC,(R2)+
848 016672  112722  000032                  MOVB    #RMOF,(R2)+
849 016676  112722  000000                  MOVB    #RMCS1,(R2)+
850 016702  112722  000200                  MOVB    #200,(R2)+
851
852 016706  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    016712  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
    016714  000240                          NOP                     ;RETURN HERE IF ERROR
    016716  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    016720  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
853 016724                          40$:
854
855                                 ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
856 016724  004737  037024                  JSR     PC,GETSTS       ;SETUP FOR STATUS
857 016730  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
858 016734                          50$:
859
860                                 ;GO READ SEEK STATUS
861 016734  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    016740  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
```

```
                016742  000240                          NOP                     ;RETURN HERE IF ERROR
                016744  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                016746  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
    862 016752                              60$:
    863
    864                                 ;VERIFY THE RESULTS OF THE SEEK COMMAND
    865 016752  004737  045224                          JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
                016756  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
                016760  000240                          NOP                     ;RETURN HERE IF ERROR
                016762  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
                016764  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                016766  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
    866 016772                              70$:
    867
    868                                 ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
    869 016772  012737  000063  001410                  MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
    870 017000  012702  001554                          MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
    871 017004  112722  000002                          MOVB    #RMWC,(R2)+
    872 017010  112722  000004                          MOVB    #RMBA,(R2)+
    873 017014  112722  000000                          MOVB    #RMCS1,(R2)+
    874 017020  112722  000200                          MOVB    #200,(R2)+
    875
    876 017024  004737  037360                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                017030  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                017032  000240                          NOP                     ;RETURN HERE IF ERROR
                017034  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                017036  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
    877 017042                              80$:
    878
    879                                 ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
    880 017042  004737  037722                          JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
    881
    882 017046  004737  037110                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                017052  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
                017054  000240                          NOP                     ;RETURN HERE IF ERROR
                017056  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                017060  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
    883 017064                              90$:
    884
    885                                 ;VERIFY RESULTS OF WRITE COMMAND
    886 017064  004737  040106                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                017070  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
                017072  000240                          NOP                     ;RETURN HERE IF ERROR
                017074  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                017076  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017100  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
    887 017104                              100$:
    888 017104  004737  052622                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                017110  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
                017112  000240                          NOP                     ;RETURN HERE IF ERROR
                017114  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                017116  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017120  000137  017272                  JMP     180$            ;GO TO 180$ IF ERROR
    889 017124                              110$:
    890 017124  004737  040740                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                017130  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
                017132  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
                017134  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                017136  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017140  000137  017272          JMP     180$            ;GO TO 180$ IF ERROR
    891         017144                  120$:
    892
    893                                 ;WRITE CHECK HEADER AND DATA FOR SECTORS JUST WRITTEN
    894         017144  012737  000053  001410  MOV     #WCH!GO,RMCS1O  ;WRITE CHECK COMMAND
    895
    896         017152  004737  037360          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                017156  000404                  BR      130$            ;GO TO 130$ IF NO ERROR
                017160  000240                  NOP                     ;RETURN HERE IF ERROR
                017162  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                017164  000137  017272          JMP     180$            ;GO TO 180$ IF ERROR
    897         017170                  130$:
    898
    899                                 ;WAIT FOR WRITE CHECK TO COMPLETE AND GET STATUS
    900         017170  004737  037722          JSR     PC,TIMOUT       ;WAIT FOR WRITE CHECK TO FINISH
    901
    902         017174  004737  037110          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                017200  000404                  BR      140$            ;GO TO 140$ IF NO ERROR
                017202  000240                  NOP                     ;RETURN HERE IF ERROR
                017204  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                017206  000137  017272          JMP     180$            ;GO TO 180$ IF ERROR
    903         017212                  140$:
    904
    905                                 ;VERIFY THE RESULTS OF WRITE CHECK OPERATION
    906         017212  004737  040106          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                017216  000405                  BR      150$            ;GO TO 150$ IF NO ERROR
                017220  000240                  NOP                     ;RETURN HERE IF ERROR
                017222  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                017224  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017226  000137  017272          JMP     180$            ;GO TO 180$ IF ERROR
    907         017232                  150$:
    908         017232  004737  052622          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                017236  000405                  BR      160$            ;GO TO 160$ IF NO ERROR
                017240  000240                  NOP                     ;RETURN HERE IF ERROR
                017242  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                017244  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017246  000137  017272          JMP     180$            ;GO TO 180$ IF ERROR
    909         017252                  160$:
    910         017252  004737  040740          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                017256  000405                  BR      170$            ;GO TO 170$ IF NO ERROR
                017260  000240                  NOP                     ;RETURN HERE IF ERROR
                017262  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                017264  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017266  000137  017272          JMP     180$            ;GO TO 180$ IF ERROR
    911         017272                  170$:
    912
    913         017272                  180$:
    914
    915                                 ;;*******************************************************************
                                        ;*TEST 14       FORMAT W/ HEAD SWITCHING
                                        ;;*******************************************************************
                017272                  TST14:
                017272  000004                  SCOPE                   ;SCOPE CALL
                017274  000240                  NOP                     ;START OF TEST
                017276  012706  001100          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
```

```
                017302  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                017306  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                017312  012737  000014  001226          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    916
    917                                         ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
    918 017320  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
    919 017326  112737  000000  001417          MOVB    #0,RMDA0+1      ;TRACK = 0
    920 017334  112737  000037  001416  5$:     MOVB    #31.,RMDA0      ;SECTOR = 31.
    921 017342  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
    922 017350  012737  176774  001412          MOV     #-258.*2,RMWC0  ;WORD COUNT FOR 2 SECTORS (2'S COMP)
    923 017356  012737  101206  001414          MOV     #BUF0NE,RMBA0   ;DATA BUFFER ADDRESS
    924 017364  012737  000062  001410          MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
    925
    926                                         ;VERIFY THAT SECTOR IS NOT BAD
                017372  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
                017376  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
                0174C^  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
                017404  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
                017406  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
    927 017412                                  10$:
    928 017412  123727  001416  000037          CMPB    RMDA0,#31.      ;IS LAST SECTOR ASSIGNED ?
    929 017420  001345                          BNE     5$              ;BR IF NO
    930 017422  012737  064620  001174          MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
    931 017430  012737  000001  001176          MOV     #1,$TMP1
    932 017436  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
    933
    934                                         ;PREPARE DEVICE FOR DATA TRANSFER
    935 017442  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                017446  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
                017450  000404                  BR      30$             ;GO TO 30$ IF NO ERROR
                017452  000240                  NOP                     ;RETURN HERE IF ERROR
                017454  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                017456  000137  020104          JMP     180$            ;GO TO 180$ IF ERROR
    936 017462                          30$:
    937
    938                                         ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
    939 017462  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
    940 017470  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
    941 017474  112722  000006                  MOVB    #RMDA,(R2)+
    942 017500  112722  000034                  MOVB    #RMDC,(R2)+
    943 017504  112722  000032                  MOVB    #RMOF,(R2)+
    944 017510  112722  000000                  MOVB    #RMCS1,(R2)+
    945 017514  112722  000200                  MOVB    #200,(R2)+
    946
    947 017520  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                017524  000404                  BR      40$             ;GO TO 40$ IF NO ERROR
                017526  000240                  NOP                     ;RETURN HERE IF ERROR
                017530  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                017532  000137  020104          JMP     180$            ;GO TO 180$ IF ERROR
    948 017536                          40$:
```

```
                                        ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
017536  004737  037024                          JSR     PC,GETSTS       ;SETUP FOR STATUS
017542  004737  037722                          JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
017546                                  50$:

                                        ;GO READ SEEK STATUS
017546  004737  037110                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
017552  000404                                  BR      60$             ;GO TO 60$ IF NO ERROR
017554  000240                                  NOP                     ;RETURN HERE IF ERROR
017556  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
017560  000137  020104                          JMP     180$            ;GO TO 180$ IF ERROR
017564                                  60$:

                                        ;VERIFY THE RESULTS OF THE SEEK COMMAND
017564  004737  045224                          JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
017570  000405                                  BR      70$             ;GO TO 70$ IF NO ERROR
017572  000240                                  NOP                     ;RETURN HERE IF ERROR
017574  104000                                  EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
017576  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
017600  000137  020104                          JMP     180$            ;GO TO 180$ IF ERROR
017604                                  70$:

                                        ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
017604  012737  000063  001410                  MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
017612  012702  001554                          MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
017616  112722  000002                          MOVB    #RMWC,(R2)+
017622  112722  000004                          MOVB    #RMBA,(R2)+
017626  112722  000000                          MOVB    #RMCSI,(R2)+
017632  112722  000200                          MOVB    #200,(R2)+

017636  004737  037360                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
017642  000404                                  BR      80$             ;GO TO 80$ IF NO ERROR
017644  000240                                  NOP                     ;RETURN HERE IF ERROR
017646  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
017650  000137  020104                          JMP     180$            ;GO TO 180$ IF ERROR
017654                                  80$:

                                        ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
017654  004737  037722                          JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

017660  004737  037110                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
017664  000404                                  BR      90$             ;GO TO 90$ IF NO ERROR
017666  000240                                  NOP                     ;RETURN HERE IF ERROR
017670  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
017672  000137  020104                          JMP     180$            ;GO TO 180$ IF ERROR
017676                                  90$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
017676  004737  040106                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
017702  000405                                  BR      100$            ;GO TO 100$ IF NO ERROR
017704  000240                                  NOP                     ;RETURN HERE IF ERROR
017706  104000                                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
017710  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
017712  000137  020104                          JMP     180$            ;GO TO 180$ IF ERROR
017716                                  100$:
017716  004737  052622                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
```

```
        017722  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        017724  000240                          NOP                     ;RETURN HERE IF ERROR
        017726  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        017730  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017732  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
   984  017736                          110$:
   985  017736  004737  040740                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        017742  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        017744  000240                          NOP                     ;RETURN HERE IF ERROR
        017746  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        017750  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017752  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
   986  017756                          120$:
   987
   988                                  ;WRITE CHECK HEADER AND DATA FOR SECTORS JUST WRITTEN
   989  017756  012737  000053  001410          MOV     #WCH!GO,RMCS10  ;WRITE CHECK COMMAND
   990
   991  017764  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        017770  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        017772  000240                          NOP                     ;RETURN HERE IF ERROR
        017774  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        017776  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
   992  020002                          130$:
   993
   994                                  ;WAIT FOR WRITE CHECK TO COMPLETE AND GET STATUS
   995  020002  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR WRITE CHECK TO FINISH
   996
   997  020006  004737  037110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        020012  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        020014  000240                          NOP                     ;RETURN HERE IF ERROR
        020016  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        020020  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
   998  020024                          140$:
   999
  1000                                  ;VERIFY THE RESULTS OF WRITE CHECK OPERATION
  1001  020024  004737  040106                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        020030  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        020032  000240                          NOP                     ;RETURN HERE IF ERROR
        020034  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        020036  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020040  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
  1002  020044                          150$:
  1003  020044  004737  052622                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        020050  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        020052  000240                          NOP                     ;RETURN HERE IF ERROR
        020054  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        020056  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020060  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
  1004  020064                          160$:
  1005  020064  004737  040740                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        020070  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
        020072  000240                          NOP                     ;RETURN HERE IF ERROR
        020074  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        020076  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020100  000137  020104                  JMP     180$            ;GO TO 180$ IF ERROR
  1006  020104                          170$:
  1007
```

```
1008 020104                                  180$:
1009
1010                                         ;;*****************************************************************
                                             ;*TEST 15       FORMAT W/ MID TRANSFER SEEK
                                             ;;*****************************************************************
     020104                                  TST15:
     020104  000004                                 SCOPE                   ;SCOPE CALL
     020106  000240                                 NOP                     ;START OF TEST
     020110  012706  001100                         MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     020114  013700  001276                         MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     020120  013701  001464                         MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     020124  012737  000015  001226                 MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1011
1012                                         ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
1013 020132  012737  000000  001444                 MOV     #0,RMDC0        ;CYLINDER = 0
1014 020140  013737  001332  001416  5$:            MOV     LSTRK,RMDA0     ;START LAST TRACK AND
1015 020146  112737  000037  001416                 MOVB    #31.,RMDA0      ;LAST SECTOR
1016 020154  012737  010000  001442                 MOV     #FMT16,RMOF0    ;16 BIT FORMAT
1017 020162  012737  176774  001412                 MOV     #-258.*2,RMWC0  ;WORD COUNT FOR 2 SECTORS (2'S COMP)
1018 020170  012737  101206  001414                 MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
1019 020176  012737  000062  001410                 MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
1020
1021                                         ;VERIFY THAT SECTOR IS NOT BAD
     020204  004737  034276                         JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     020210  000405                                 BR      10$             ;GO TO 10$ IF NO ERROR
     020212  104401  063120                         TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     020216  104000                                 EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     020220  000137  020716                         JMP     180$            ;GO TO 180$ IF ERROR
1022 020224                                  10$:
1023 020224  123737  001417  001333                 CMPB    RMDA0+1,LSTRK+1 ;IS LAST TRACK ASSIGNED ?
1024 020232  001342                                 BNE     5$              ;BR IF NO
1025 020234  012737  064620  001174                 MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
1026 020242  012737  000001  001176                 MOV     #1,$TMP1
1027 020250  004737  036224                         JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
1028 020254                                  20$:
1029
1030                                         ;PREPARE DEVICE FOR DATA TRANSFER
1031 020254  004737  033352                         JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     020260  154130                                 .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                             ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                             ;CLEAR CONTROLLER & SELECT DEVICE
                                                                             ;VERIFY CONTROLLER CLEAR OPERATION
                                                                             ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                             ;VERIFY PACK ACKNOWLEDGE
                                                                             ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                             ;VERIFY RECALIBRATION
     020262  000404                                 BR      30$             ;GO TO 30$ IF NO ERROR
     020264  000240                                 NOP                     ;RETURN HERE IF ERROR
     020266  104000                                 EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     020270  000137  020716                         JMP     180$            ;GO TO 180$ IF ERROR
1032 020274                                  30$:
1033
1034                                         ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE ON CYLINDER
1035 020274  012737  000005  001410                 MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
1036 020302  012702  001551                         MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1037 020306  112722  000006                         MOVB    #RMDA,(R2)+
1038 020312  112722  000034                         MOVB    #RMDC,(R2)+
```

```
1039 020316  112722  000032                MOVB    #RMOF,(R2)+
1040 020322  112722  000000                MOVB    #RMCS1,(R2)+
1041 020326  112722  000200                MOVB    #200,(R2)+
1042
1043 020332  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020336  000404                        BR      40$             ;GO TO 40$ IF NO ERROR
     020340  000240                        NOP                     ;RETURN HERE IF ERROR
     020342  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020344  000137  020716                JMP     180$            ;GO TO 180$ IF ERROR
1044 020350                          40$:
1045
1046                                 ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
1047 020350  004737  037024                JSR     PC,GETSTS       ;SETUP FOR STATUS
1048 020354  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
1049 020360                          50$:
1050
1051
1052 020360  004737  037110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020364  000404                        BR      60$             ;GO TO 60$ IF NO ERROR
     020366  000240                        NOP                     ;RETURN HERE IF ERROR
     020370  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020372  000137  020716                JMP     180$            ;GO TO 180$ IF ERROR
1053 020376                          60$:
1054
1055                                 ;VERIFY THE RESULTS OF THE SEEK COMMAND
1056 020376  004737  045224                JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     020402  000405                        BR      70$             ;GO TO 70$ IF NO ERROR
     020404  000240                        NOP                     ;RETURN HERE IF ERROR
     020406  104000                        EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     020410  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020412  000137  020716                JMP     180$            ;GO TO 180$ IF ERROR
1057 020416                          70$:
1058
1059                                 ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
1060 020416  012737  000063  001410        MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
1061 020424  012702  001554                MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
1062 020430  112722  000002                MOVB    #RMWC,(R2)+
1063 020434  112722  000004                MOVB    #RMBA,(R2)+
1064 020440  112722  000000                MOVB    #RMCS1,(R2)+
1065 020444  112722  000200                MOVB    #200,(R2)+
1066
1067 020450  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020454  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
     020456  000240                        NOP                     ;RETURN HERE IF ERROR
     020460  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020462  000137  020716                JMP     180$            ;GO TO 180$ IF ERROR
1068 020466                          80$:
1069
1070                                 ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
1071 020466  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1072
1073 020472  004737  037110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020476  000404                        BR      90$             ;GO TO 90$ IF NO ERROR
     020500  000240                        NOP                     ;RETURN HERE IF ERROR
     020502  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020504  000137  020716                JMP     180$            ;GO TO 180$ IF ERROR
1074 020510                          90$:
```

```
1075
1076                                    ;VERIFY RESULTS OF WRITE COMMAND
1077 020510  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     020514  000405                              BR      100$            ;GO TO 100$ IF NO ERROR
     020516  000240                              NOP                     ;RETURN HERE IF ERROR
     020520  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     020522  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020524  000137  020716                      JMP     180$            ;GO TO 180$ IF ERROR
1078 020530                             100$:
1079 020530  004737  052622                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     020534  000405                              BR      110$            ;GO TO 110$ IF NO ERROR
     020536  000240                              NOP                     ;RETURN HERE IF ERROR
     020540  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     020542  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020544  000137  020716                      JMP     180$            ;GO TO 180$ IF ERROR
1080 020550                             110$:
1081 020550  004737  040740                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     020554  000405                              BR      120$            ;GO TO 120$ IF NO ERROR
     020556  000240                              NOP                     ;RETURN HERE IF ERROR
     020560  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     020562  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020564  000137  020716                      JMP     180$            ;GO TO 180$ IF ERROR
1082 020570                             120$:
1083
1084                                    ;WRITE CHECK HEADER AND DATA FOR SECTORS JUST WRITTEN
1085 020570  012737  000053  001410              MOV     #WCH!GO,RMCS10  ;WRITE CHECK COMMAND
1086
1087 020576  004737  037360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020602  000404                              BR      130$            ;GO TO 130$ IF NO ERROR
     020604  000240                              NOP                     ;RETURN HERE IF ERROR
     020606  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020610  000137  020716                      JMP     180$            ;GO TO 180$ IF ERROR
1088 020614                             130$:
1089
1090                                    ;WAIT FOR WRITE CHECK TO COMPLETE AND GET STATUS
1091 020614  004737  037722                      JSR     PC,TIMOUT       ;WAIT FOR WRITE CHECK TO FINISH
1092
1093 020620  004737  037110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020624  000404                              BR      140$            ;GO TO 140$ IF NO ERROR
     020626  000240                              NOP                     ;RETURN HERE IF ERROR
     020630  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020632  000137  020716                      JMP     180$            ;GO TO 180$ IF ERROR
1094 020636                             140$:
1095
1096                                    ;VERIFY THE RESULTS OF WRITE CHECK OPERATION
1097 020636  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     020642  000405                              BR      150$            ;GO TO 150$ IF NO ERROR
     020644  000240                              NOP                     ;RETURN HERE IF ERROR
     020646  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     020650  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020652  000137  020716                      JMP     180$            ;GO TO 180$ IF ERROR
1098 020656                             150$:
1099 020656  004737  052622                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     020662  000405                              BR      160$            ;GO TO 160$ IF NO ERROR
     020664  000240                              NOP                     ;RETURN HERE IF ERROR
     020666  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     020670  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
                020672  000137  020716                  JMP     180$            ;GO TO 180$ IF ERROR
     1100       020676                          160$:
     1101       020676  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                020702  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
                020704  000240                          NOP                     ;RETURN HERE IF ERROR
                020706  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                020710  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                020712  000137  020716                  JMP     180$            ;GO TO 180$ IF ERROR
     1102       020716                          170$:
     1103
     1104       020716                          180$:
     1105
     1106                                       ;;*****************************************************************
                                                ;*TEST 16        FORMAT W/ IMPLIED SEEK
                                                ;;*****************************************************************
                020716                          TST16:
                020716  000004                          SCOPE                   ;SCOPE CALL
                020720  000240                          NOP                     ;START OF TEST
                020722  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                020726  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                020732  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                020736  012737  000016  001226          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     1107
     1108                                       ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
     1109       020744  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
     1110       020752  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
     1111       020760  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
     1112       020766  012737  176774  001412          MOV     #-258.*2,RMWC0  ;WORD COUNT FOR 2 SECTORS (2'S COMP)
     1113       020774  012737  101206  001414          MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
     1114       021002  012737  000062  001410          MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
     1115
     1116                                       ;VERIFY THAT SECTOR IS NOT BAD
                021010  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
                021014  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
                021016  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
                021022  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
                021024  000137  021534                  JMP     180$            ;GO TO 180$ IF ERROR
     1117       021030                          10$:
     1118       021030  012737  064620  001174          MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
     1119       021036  012737  000001  001176          MOV     #1,$TMP1
     1120       021044  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
     1121       021050                          20$:
     1122
     1123                                       ;PREPARE DEVICE FOR DATA TRANSFER
     1124       021050  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                021054  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
                021056  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
                021060  000240                          NOP                     ;RETURN HERE IF ERROR
                021062  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                021064  000137  021534                  JMP     180$            ;GO TO 180$ IF ERROR
```

```
1125 021070                                30$:
1126
1127
1128 021070  013737  001444  021536        ;SETUP PARAMETERS AND EXECUTE SEEK TO GET DRIVE OFF CYLINDER
1129 021076  012737  001466  001444              MOV     RMDCO,190$      ;SAVE CYLINDER ADDRESS
1130 021104  012737  000005  001410              MOV     #822.,RMDCO     ;SEEK TO LAST CYLINDER
1131 021112  012702  001551                      MOV     #SEEK!GO,RMCS10 ;CHANGE COMMAND TO SEEK
1132 021116  112722  000006                      MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1133 021122  112722  000034                      MOVB    #RMDA,(R2)+
1134 021126  112722  000032                      MOVB    #RMDC,(R2)+
1135 021132  112722  000000                      MOVB    #RMOF,(R2)+
1136 021136  112722  000200                      MOVB    #RMCS1,(R2)+
1137                                            MOVB    #200,(R2)+
1138 021142  004737  037360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021146  000404                              BR      40$             ;GO TO 40$ IF NO ERROR
     021150  000240                              NOP                     ;RETURN HERE IF ERROR
     021152  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021154  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1139 021160                                40$:
1140
1141
1142 021160  004737  037024        ;SETUP FOR READING STATUS AND THEN WAIT FOR SEEK TO COMPLETE
1143 021164  004737  037722              JSR     PC,GETSTS       ;SETUP FOR STATUS
1144 021170                              JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
1145                                    50$:
1146
1147 021170  004737  037110        ;GO READ SEEK STATUS
     021174  000404                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021176  000240                      BR      60$             ;GO TO 60$ IF NO ERROR
     021200  104000                      NOP                     ;RETURN HERE IF ERROR
     021202  000137  021534              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
1148 021206                              JMP     180$            ;GO TO 180$ IF ERROR
1149                                    60$:
1150
1151 021206  004737  045224        ;VERIFY THE RESULTS OF THE SEEK COMMAND
     021212  000405                      JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     021214  000240                      BR      70$             ;GO TO 70$ IF NO ERROR
     021216  104000                      NOP                     ;RETURN HERE IF ERROR
     021220  004736                      EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     021222  000137  021534              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1152 021226                              JMP     180$            ;GO TO 180$ IF ERROR
1153                                    70$:
1154
1155 021226  013737  021536  001444  ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
1156 021234  012737  000063  001410        MOV     190$,RMDCO      ;RESTORE DISK ADDRESS
1157 021242  012702  001554              MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
1158 021246  112722  000002              MOV     #PUTINX+3,R2    ;EXTEND REGISTER INDEX TABLE
1159 021252  112722  000004              MOVB    #RMWC,(R2)+
1160 021256  112722  000000              MOVB    #RMBA,(R2)+
1161 021262  112722  000200              MOVB    #RMCS1,(R2)+
1162                                    MOVB    #200,(R2)+
1163 021266  004737  037360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021272  000404                      BR      80$             ;GO TO 80$ IF NO ERROR
     021274  000240                      NOP                     ;RETURN HERE IF ERROR
     021276  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021300  000137  021534              JMP     180$            ;GO TO 180$ IF ERROR
1164 021304                        80$.
```

```
1165
1166                                    ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
1167 021304  004737  037722                      JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1168
1169 021310  004737  037110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021314  000404                              BR      90$             ;GO TO 90$ IF NO ERROR
     021316  000240                              NOP                     ;RETURN HERE IF ERROR
     021320  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021322  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1170 021326                             90$:
1171
1172                                    ;VERIFY RESULTS OF WRITE COMMAND
1173 021326  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     021332  000405                              BR      100$            ;GO TO 100$ IF NO ERROR
     021334  000240                              NOP                     ;RETURN HERE IF ERROR
     021336  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     021340  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021342  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1174 021346                             100$:
1175 021346  004737  052622                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     021352  000405                              BR      110$            ;GO TO 110$ IF NO ERROR
     021354  000240                              NOP                     ;RETURN HERE IF ERROR
     021356  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     021360  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021362  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1176 021366                             110$:
1177 021366  004737  040740                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     021372  000405                              BR      120$            ;GO TO 120$ IF NO ERROR
     021374  000240                              NOP                     ;RETURN HERE IF ERROR
     021376  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     021400  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021402  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1178 021406                             120$:
1179
1180                                    ;WRITE CHECK HEADER AND DATA FOR SECTORS JUST WRITTEN
1181 021406  012737  000053  001410              MOV     #WCH!GO,RMCS10  ;WRITE CHECK COMMAND
1182
1183 021414  004737  037360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021420  000404                              BR      130$            ;GO TO 130$ IF NO ERROR
     021422  000240                              NOP                     ;RETURN HERE IF ERROR
     021424  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021426  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1184 021432                             130$:
1185
1186                                    ;WAIT FOR WRITE CHECK TO COMPLETE AND GET STATUS
1187 021432  004737  037722                      JSR     PC,TIMOUT       ;WAIT FOR WRITE CHECK TO FINISH
1188
1189 021436  004737  037110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021442  000404                              BR      140$            ;GO TO 140$ IF NO ERROR
     021444  000240                              NOP                     ;RETURN HERE IF ERROR
     021446  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021450  000137  021534                      JMP     180$            ;GO TO 180$ IF ERROR
1190 021454                             140$:
1191
1192                                    ;VERIFY THE RESULTS OF WRITE CHECK OPERATION
1193 021454  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     021460  000405                              BR      150$            ;GO TO 150$ IF NO ERROR
```

```
                021462  000240                          NOP                     ;RETURN HERE IF ERROR
                021464  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                021466  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                021470  000137  021534                  JMP     180$            ;GO TO 180$ IF ERROR
1194            021474                          150$:
1195            021474  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                021500  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
                021502  000240                          NOP                     ;RETURN HERE IF ERROR
                021504  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                021506  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                021510  000137  021534                  JMP     180$            ;GO TO 180$ IF ERROR
1196            021514                          160$:
1197            021514  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                021520  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
                021522  000240                          NOP                     ;RETURN HERE IF ERROR
                021524  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                021526  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                021530  000137  021534                  JMP     180$            ;GO TO 180$ IF ERROR
1198            021534                          170$:
1199
1200            021534  000401                  180$:   BR      200$
1201
1202            021536  000000                  190$:   .WORD   0               ;TEMPORARY STORAGE
1203
1204            021540                          200$:
1205
1206                                            ;;*****************************************************************
                                                ;*TEST 17       FORMAT EACH SECTOR ADDRESS
                                                ;;*****************************************************************
                021540                          TST17:
                021540  000004                          SCOPE                   ;SCOPE CALL
                021542  000240                          NOP                     ;START OF TEST
                021544  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                021550  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                021554  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                021560  012737  000017  001226          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1207
1208                                            ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
1209            021566  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
1210            021574  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
1211            021602  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
1212            021610                          5$:
1213            021610  012737  177376  001412          MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
1214            021616  012737  101206  001414          MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
1215            021624  012737  000062  001410          MOV     #WH,RMCS1       ;WRITE HEADER AND DATA
1216
1217                                            ;VERIFY THAT SECTOR IS NOT BAD
                021632  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
                021636  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
                021640  104401  063120                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
                021644  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
                021646  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1218            021652                          10$:
1219            021652  012737  001416  001174          MOV     #RMDA0,$TMP0    ;USE SECTOR FOR DATA PATTERN
1220            021660  012737  000001  001176          MOV     #1,$TMP1
1221            021666  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
1222            021672                          20$:
```

```
                                        ;PREPARE DEVICE FOR DATA TRANSFER
021672  004737  033352                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
021676  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
021700  000404                              BR      30$             ;GO TO 30$ IF NO ERROR
021702  000240                              NOP                     ;RETURN HERE IF ERROR
021704  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
021706  000137  022302                      JMP     190$            ;GO TO 190$ IF ERROR
021712                                  30$:

                                        ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
021712  012737  000063  001410              MOV     #WH!GO,RMCS1O   ;WRITE HEADER AND DATA
021720  012702  001551                      MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
021724  112722  000006                      MOVB    #RMDA,(R2)+
021730  112722  000034                      MOVB    #RMDC,(R2)+
021734  112722  000032                      MOVB    #RMOF,(R2)+
021740  112722  000002                      MOVB    #RMWC,(R2)+
021744  112722  000004                      MOVB    #RMBA,(R2)+
021750  112722  000000                      MOVB    #RMCS1,(R2)+
021754  112722  000200                      MOVB    #200,(R2)+

021760  004737  037360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
021764  000404                              BR      80$             ;GO TO 80$ IF NO ERROR
021766  000240                              NOP                     ;RETURN HERE IF ERROR
021770  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
021772  000137  022302                      JMP     190$            ;GO TO 190$ IF ERROR
021776                                  80$:

                                        ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
021776  004737  037024                      JSR     PC,GETSTS
022002  004737  037722                      JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

022006  004737  037110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
022012  000404                              BR      90$             ;GO TO 90$ IF NO ERROR
022014  000240                              NOP                     ;RETURN HERE IF ERROR
022016  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
022020  000137  022302                      JMP     190$            ;GO TO 190$ IF ERROR
022024                                  90$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
022024  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
022030  000405                              BR      100$            ;GO TO 100$ IF NO ERROR
022032  000240                              NOP                     ;RETURN HERE IF ERROR
022034  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
022036  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
022040  000137  022302                      JMP     190$            ;GO TO 190$ IF ERROR
022044                                  100$:
022044  004737  052622                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
022050  000405                              BR      110$            ;GO TO 110$ IF NO ERROR
022052  000240                              NOP                     ;RETURN HERE IF ERROR
```

```
                022054  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                022056  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022060  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1253            022064                          110$:
1254            022064  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                022070  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
                022072  000240                          NOP                     ;RETURN HERE IF ERROR
                022074  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                022076  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022100  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1255            022104                          120$:
1256
1257                                            ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
1258            022104  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
1259            022112  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
1260
1261            022120  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                022124  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
                022126  000240                          NOP                     ;RETURN HERE IF ERROR
                022130  104000                          EMT                     ;ERROR # DEFINFD BY PUT SUBROUTINE
                022132  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1262            022136                          130$:
1263
1264                                            ;WAIT FOR READ TO COMPLETE AND GET STATUS
1265            022136  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
1266
1267            022142  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                022146  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
                022150  000240                          NOP                     ;RETURN HERE IF ERROR
                022152  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                022154  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1268            022160                          140$:
1269
1270                                            ;VERIFY THE RESULTS OF READ OPERATION
1271            022160  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                022164  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                022166  000240                          NOP                     ;RETURN HERE IF ERROR
                022170  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                022172  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022174  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1272            022200                          150$:
1273            022200  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                022204  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
                022206  000240                          NOP                     ;RETURN HERE IF ERROR
                022210  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                022212  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022214  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1274            022220                          160$:
1275            022220  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                022224  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
                022226  000240                          NOP                     ;RETURN HERE IF ERROR
                022230  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                022232  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022234  000137  022302                  JMP     190$            ;GO TO 190$ IF ERROR
1276            022240                          170$:
1277
1278                                            ;VERIFY DATA
```

```
 1279 022240  004737  036462                JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
      022244  101206                        .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
      022246  102212                        .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
      022250  000404                        BR      180$            ;GO TO 180$ IF NO ERROR
      022252  000240                        NOP                     ;RETURN HERE IF ERROR
      022254  104000                        EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
      022256  000137  022302                JMP     190$            ;GO TO 190$ IF ERROR
 1280
 1281                                ;INCREMENT ADDRESS AND FORMAT NEXT SECTOR
 1282 022262                         180$:
 1283 022262  005237  001416                INC     RMDAO           ;ADVANCE SECTOR COUNT
 1284 022266  122737  000037  001416        CMPB    #31.,RMDAO      ;DONE ALL SECTORS??
 1285 022274  103402                        BLO     190$            ;YES!!
 1286 022276  000137  021610                JMP     5$              ;GO DO NEXT SECTOR
 1287 022302                         190$:
 1288
 1289                                ;;*****************************************************************
                                     ;*TEST 20       FORMAT EACH TRACK ADDRESS
                                     ;;*****************************************************************
      022302                         TST20:
      022302  000004                        SCOPE                   ;SCOPE CALL
      022304  000240                        NOP                     ;START OF TEST
      022306  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      022312  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      022316  013701  001464                MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      022322  012737  000020  001226        MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1290
 1291                                ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
 1292 022330  012737  000000  001444        MOV     #0,RMDC0        ;CYLINDER = 0
 1293 022336  012737  000000  001416        MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
 1294 022344  012737  010000  001442        MOV     #FMT16,RMOF0    ;16 BIT FORMAT
 1295 022352                         5$:
 1296 022352  012737  177376  001412        MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
 1297 022360  012737  101206  001414        MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
 1298 022366  012737  000062  001410        MOV     #WH,RMCS1       ;WRITE HEADER AND DATA
 1299
 1300                                ;VERIFY THAT SECTOR IS NOT BAD
      022374  004737  034276                JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
      022400  000405                        BR      10$             ;GO TO 10$ IF NO ERROR
      022402  104401  063120                TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
      022406  104000                        EMT                     ;ERROR # DEFINED BY BADSCT SUBRO TINE
      022410  000137  023044                JMP     190$            ;GO TO 190$ IF ERROR
 1301 022414                         10$:
 1302 022414  012737  001416  001174        MOV     #RMDA0,$TMP0    ;USE TRACK FOR DATA PATTERN
 1303 022422  012737  000001  001176        MOV     #1,$TMP1
 1304 022430  004737  036224                JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
 1305 022434                         20$:
 1306
 1307                                ;PREPARE DEVICE FOR DATA TRANSFER
 1308 022434  004737  033352                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      022440  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
```

```
                022442  000404                          BR      30$             ;VERIFY RECALIBRATION
                022444  000240                          NOP                     ;GO TO 30$ IF NO ERROR
                022446  104000                          EMT                     ;RETURN HERE IF ERROR
                022450  000137  023044                  JMP     190$            ;ERROR # DEFINED BY TSTPRP SUBROUTINE
1309            022454                          30$:                            ;GO TO 190$ IF ERROR
1310
1311                                            ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
1312            022454  012737  000063  001410          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
1313            022462  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1314            022466  112722  000006                  MOVB    #RMDA,(R2)+
1315            022472  112722  000034                  MOVB    #RMDC,(R2)+
1316            022476  112722  000032                  MOVB    #RMOF,(R2)+
1317            022502  112722  000002                  MOVB    #RMWC,(R2)+
1318            022506  112722  000004                  MOVB    #RMBA,(R2)+
1319            022512  112722  000000                  MOVB    #RMCS1,(R2)+
1320            022516  112722  000200                  MOVB    #200,(R2)+
1321
1322            022522  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                022526  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                022530  000240                          NOP                     ;RETURN HERE IF ERROR
                022532  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                022534  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1323            022540                          80$:
1324
1325                                            ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1326            022540  004737  037024                  JSR     PC,GETSTS
1327            022544  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1328
1329            022550  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                022554  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
                022556  000240                          NOP                     ;RETURN HERE IF ERROR
                022560  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                022562  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1330            022566                          90$:
1331
1332                                            ;VERIFY RESULTS OF WRITE COMMAND
1333            022566  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                022572  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
                022574  000240                          NOP                     ;RETURN HERE IF ERROR
                022576  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                022600  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022602  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1334            022606                          100$:
1335            022606  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                022612  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
                022614  000240                          NOP                     ;RETURN HERE IF ERROR
                022616  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                022620  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022622  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1336            022626                          110$:
1337            022626  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                022632  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
                022634  000240                          NOP                     ;RETURN HERE IF ERROR
                022636  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                022640  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022642  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
```

```
1338 022646                          120$:
1339
1340                                 ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
1341 022646  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
1342 022654  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
1343
1344 022662  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022666  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
     022670  000240                          NOP                     ;RETURN HERE IF ERROR
     022672  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     022674  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1345 022700                          130$:
1346
1347                                 ;WAIT FOR READ TO COMPLETE AND GET STATUS
1348 022700  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
1349
1350 022704  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022710  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
     022712  000240                          NOP                     ;RETURN HERE IF ERROR
     022714  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     022716  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1351 022722                          140$:
1352
1353                                 ;VERIFY THE RESULTS OF READ OPERATION
1354 022722  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     022726  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
     022730  000240                          NOP                     ;RETURN HERE IF ERROR
     022732  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     022734  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022736  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1355 022742                          150$:
1356 022742  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     022746  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
     022750  000240                          NOP                     ;RETURN HERE IF ERROR
     022752  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     022754  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022756  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1357 022762                          160$:
1358 022762  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     022766  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
     022770  000240                          NOP                     ;RETURN HERE IF ERROR
     022772  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     022774  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022776  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1359 023002                          170$:
1360
1361                                 ;VERIFY DATA
1362 023002  004737  036462                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
     023006  101206                          .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
     023010  102212                          .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
     023012  000404                          BR      180$            ;GO TO 180$ IF NO ERROR
     023014  000240                          NOP                     ;RETURN HERE IF ERROR
     023016  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
     023020  000137  023044                  JMP     190$            ;GO TO 190$ IF ERROR
1363
1364                                 ;INCREMENT ADDRESS AND FORMAT NEXT TRACK
1365 023024                          180$:
```

```
 1366 023024  105237  001417                        INCB    RMDA0+1          ;ADVANCE TRACK COUNT
 1367 023030  123737  001417  001333                CMPB    RMDA0+1,LSTRK+1  ;LAST TRACK ?
 1368 023036  101002                                BHI     190$             ;YES!!
 1369 023040  000137  022352                        JMP     5$               ;GO DO NEXT SECTOR
 1370 023044                                190$:
 1371
 1372                                       ;;*****************************************************************
                                            ;*TEST 21       FORMAT PRIME CYLINDERS
                                            ;;*****************************************************************
      023044                                TST21:
      023044  000004                            SCOPE                    ;SCOPE CALL
      023046  000240                            NOP                      ;START OF TEST
      023050  012706  001100                    MOV     #STACK,SP        ;INITIALIZE STACK POINTER
      023054  013700  001276                    MOV     $BASE,R0         ;R0 = UNIBUS ADDRESS
      023060  013701  001464                    MOV     TSTQUE,R1        ;(R1) = DEVICE BEING TESTED
      023064  012737  000021  001226            MOV     #21,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 1373
 1374                                       ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
 1375 023072  012737  000001  001444            MOV     #1,RMDC0         ;CYLINDER = 1
 1376 023100  012737  000000  001416            MOV     #0,RMDA0         ;TRACK = 0, SECTOR = 0
 1377 023106                                5$:
 1378 023106  012737  010000  001442            MOV     #FMT16,RMOF0     ;16 BIT FORMAT
 1379 023114  012737  177376  001412            MOV     #-258.,RMWC0     ;2 + 256 WORDS (2'S COMP)
 1380 023122  012737  101206  001414            MOV     #BUFONE,RMBA0    ;DATA BUFFER ADDRESS
 1381 023130  012737  000062  001410            MOV     #WH,RMCS10       ;WRITE HEADER AND DATA
 1382
 1383                                       ;VERIFY THAT SECTOR IS NOT BAD
      023136  004737  034276                    JSR     PC,BADSCT        ;CALL BAD SECTOR MODULE
      023142  000405                            BR      10$              ;GO TO 10$ IF NO ERROR
      023144  104401  063120                    TYPE    ,SCTMSG          ;TYPE BAD SECTOR MESSAGE
      023150  104000                            EMT                      ;ERROR # DEFINED BY BADSCT SUBROUTINE
      023152  000137  023606                    JMP     190$             ;GO TO 190$ IF ERROR
 1384 023156                                10$:
 1385 023156  012737  001444  001174            MOV     #RMDC0,$TMP0     ;USE CYLINDER FOR DATA PATTERN
 1386 023164  012737  000001  001176            MOV     #1,$TMP1
 1387 023172  004737  036224                    JSR     PC,GENBUF        ;GO GENERATE DATA BUFFER
 1388
 1389 023176                                20$:
 1390
 1391                                       ;PREPARE DEVICE FOR DATA TRANSFER
 1392 023176  004737  033352                    JSR     PC,TSTPRP        ;PREPARE DEVICE FOR TEST
      023202  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
      023204  000404                            BR      30$              ;GO TO 30$ IF NO ERROR
      023206  000240                            NOP                      ;RETURN HERE IF ERROR
      023210  104000                            EMT                      ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      023212  000137  023606                    JMP     190$             ;GO TO 190$ IF ERROR
 1393 023216                                30$:
 1394
 1395                                       ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
 1396 023216  012737  000063  001410            MOV     #WH!GO,RMCS10    ;WRITE HEADER AND DATA
```

```
1397 023224  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1398 023230  112722  000006                  MOVB    #RMDA,(R2)+
1399 023234  112722  000034                  MOVB    #RMDC,(R2)+
1400 023240  112722  000032                  MOVB    #RMOF,(R2)+
1401 023244  112722  000002                  MOVB    #RMWC,(R2)+
1402 023250  112722  000004                  MOVB    #RMBA,(R2)+
1403 023254  112722  000000                  MOVB    #RMCS1,(R2)+
1404 023260  112722  000200                  MOVB    #200,(R2)+
1405 023264  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     023270  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
     023272  000240                          NOP                     ;RETURN HERE IF ERROR
     023274  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     023276  000137  023606                  JMP     190$            ; 0 TO 190$ IF ERROR
1406 023302                          80$:
1407
1408                                 ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1409 023302  004737  037024                  JSR     PC,GETSTS
1410
1411                                 ;WAIT FOR WRITE COMMAND TO COMPLETE AND READ STATUS
1412 023306  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1413 023312  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     023316  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
     023320  000240                          NOP                     ;RETURN HERE IF ERROR
     023322  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     023324  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
1414 023330                          90$:
1415
1416                                 ;VERIFY RESULTS OF WRITE COMMAND
1417 023330  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     023334  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
     023336  000240                          NOP                     ;RETURN HERE IF ERROR
     023340  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     023342  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023344  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
1418 023350                          100$:
1419 023350  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     023354  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
     023356  000240                          NOP                     ;RETURN HERE IF ERROR
     023360  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     023362  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023364  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
1420 023370                          110$:
1421 023370  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     023374  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
     023376  000240                          NOP                     ;RETURN HERE IF ERROR
     023400  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     023402  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023404  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
1422 023410                          120$:
1423
1424                                 ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
1425 023410  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
1426 023416  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
1427 023424  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     023430  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
     023432  000240                          NOP                     ;RETURN HERE IF ERROR
     023434  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
```

```
        023436  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
   1428 023442                          130$:
   1429
   1430                                 ;WAIT FOR READ TO COMPLETE AND GET STATUS
   1431 023442  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
   1432 023446  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        023452  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        023454  000240                          NOP                     ;RETURN HERE IF ERROR
        023456  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        023460  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
   1433 023464                          140$:
   1434
   1435                                 ;VERIFY THE RESULTS OF READ OPERATION
   1436 023464  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        023470  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        023472  000240                          NOP                     ;RETURN HERE IF ERROR
        023474  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        023476  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023500  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
   1437 023504                          150$:
   1438 023504  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        023510  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        023512  000240                          NOP                     ;RETURN HERE IF ERROR
        023514  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        023516  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023520  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
   1439 023524                          160$:
   1440 023524  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        023530  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
        023532  000240                          NOP                     ;RETURN HERE IF ERROR
        023534  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        023536  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023540  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
   1441 023544                          170$:
   1442
   1443                                 ;VERIFY DATA
   1444 023544  004737  036462                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        023550  101206                          .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
        023552  102212                          .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
        023554  000404                          BR      180$            ;GO TO 180$ IF NO ERROR
        023556  000240                          NOP                     ;RETURN HERE IF ERROR
        023560  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        023562  000137  023606                  JMP     190$            ;GO TO 190$ IF ERROR
   1445 023566                          180$:
   1446
   1447                                 ;INCREMENT ADDRESS AND FORMAT NEXT PRIME CYLINDER
   1448 023566  006337  001444                  ASL     RMDCO           ;ADVANCE CYLINDER COUNT
   1449 023572  023727  001444  001000          CMP     RMDCO,#512.     ;DONE ALL PRIME CYLINDERS ?
   1450 023600  003002                          BGT     190$            ;YES!!
   1451 023602  000137  023106                  JMP     5$              ;GO DO NEXT CYLINDER
   1452 023606                          190$:
   1453
   1454                                 ;;*******************************************************************
                                        ;*TEST 22        READ HEADER & DATA IN LAST SECTOR
                                        ;;*******************************************************************
        023606                          TST22:
        023606  000004                          SCOPE                   ;SCOPE CALL
```

```
        023610  000240                          NOP                     ;START OF TEST
        023612  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        023616  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        023622  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        023626  012737  000022  001226          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  1455
  1456                                  ;SETUP PARAMETERS FOR READING LAST SECTOR
  1457  023634  012737  001466  001444          MOV     #822.,RMDC0     ;LAST CYLINDER
  1458  023642  013737  001332  001416          MOV     LSTRK,RMDA0     ;SET LAST TRACK AND
  1459  023650  112737  000036  001416          MOVB    #30.,RMDA0      ;LAST SECTOR
  1460  023656  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
  1461  023664                                20$:
  1462  023664  012737  177376  001412          MOV     #-258.,RMWC0    ;2 * 256 WORDS (2'S COMP)
  1463  023672  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;DATA BUFFER ADDRESS
  1464  023700  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER AND DATA
  1465
  1466                                  ;PREPARE DEVICE FOR DATA TRANSFER
  1467  023706  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        023712  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        023714  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        023716  000240                          NOP                     ;RETURN HERE IF ERROR
        023720  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        023722  000137  024112                  JMP     190$            ;GO TO 190$ IF ERROR
  1468  023726                                30$:
  1469  023726  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
  1470  023732  112722  000006                  MOVB    #RMDA,(R2)+
  1471  023736  112722  000034                  MOVB    #RMDC,(R2)+
  1472  023742  112722  000032                  MOVB    #RMOF,(R2)+
  1473  023746  112722  000004                  MOVB    #RMBA,(R2)+
  1474  023752  112722  000002                  MOVB    #RMWC,(R2)+
  1475  023756  112722  000000                  MOVB    #RMCS1,(R2)+
  1476  023762  112722  000200                  MOVB    #200,(R2)+      ;TERMINATOR
  1477  023766                                120$:
  1478
  1479                                  ;READ HEADER AND DATA OF LAST SECTOR
  1480  023766  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        023772  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        023774  000240                          NOP                     ;RETURN HERE IF ERROR
        023776  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        024000  000137  024112                  JMP     190$            ;GO TO 190$ IF ERROR
  1481  024004                                130$:
  1482
  1483                                  ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
  1484  024004  004737  037024                  JSR     PC,GETSTS
  1485  024010  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
  1486
  1487  024014  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        024020  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        024022  000240                          NOP                     ;RETURN HERE IF ERROR
        024024  10400C                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
```

```
024026  000137  024112                  JMP     190$            ;GO TO 190$ IF ERROR
1488 024032                     140$:
1489
1490                            ;VERIFY THE RESULTS OF READ OPERATION
1491 024032  004737  040106              JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
024036  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
024040  000240                          NOP                     ;RETURN HERE IF ERROR
024042  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
024044  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
024046  000137  024112                  JMP     190$            ;GO TO 190$ IF ERROR
1492 024052                     150$:
1493 024052  004737  052622              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
024056  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
024060  000240                          NOP                     ;RETURN HERE IF ERROR
024062  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
024064  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
024066  000137  024112                  JMP     190$            ;GO TO 190$ IF ERROR
1494 024072                     160$:
1495 024072  004737  040740              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
024076  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
024100  000240                          NOP                     ;RETURN HERE IF ERROR
024102  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
024104  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
024106  000137  024112                  JMP     190$            ;GO TO 190$ IF ERROR
1496 024112                     170$:
1497
1498 024112                     190$:
1499
1500                            ;;*****************************************************************
024112                          ;*TEST 23        READ HEADER & DATA W/ AOE ERROR
                                ;;*****************************************************************
024112  000004                TST23:  SCOPE                   ;SCOPE CALL
024114  000240                          NOP                     ;START OF TEST
024116  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
024122  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
024126  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
024132  012737  000023  001226          MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1501
1502                            ;SETUP PARAMETERS FOR READING 2 SECTORS STARTING WITH LAST SECTOR
1503 024140  013737  001332  001416      MOV     LSTRK,RMDA0     ;SET LAST TRACK AND
1504 024146  112737  000036  001416      MOVB    #30.,RMDA0      ;LAST SECTOR
1505 024154  012737  010000  001442      MOV     #FMT16,RMOF0    ;16 BIT FORMAT
1506 024162                     20$:
1507 024162  012737  001466  001444      MOV     #822.,RMDC0     ;LAST CYLINDER
1508 024170  012737  176774  001412      MOV     #-258.*2,RMWC0  ;READ 2 SECTORS
1509 024176  012737  102212  001414      MOV     #BUFTWO,RMBA0   ;DATA BUFFER ADDRESS
1510 024204  012737  000073  001410      MOV     #RH!GO,RMCS1    ;READ HEADER AND DATA
1511
1512                            ;PREPARE DEVICE FOR DATA TRANSFER
1513 024212  004737  033352              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
024216  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
        024220  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        024222  000240                          NOP                     ;RETURN HERE IF ERROR
        024224  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        024226  000137  024416                  JMP     190$            ;GO TO 190$ IF ERROR
1514    024232                          30$:
1515    024232  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1516    024236  112722  000006                  MOVB    #RMDA,(R2)+
1517    024242  112722  000034                  MOVB    #RMDC,(R2)+
1518    024246  112722  000032                  MOVB    #RMOF,(R2)+
1519    024252  112722  000002                  MOVB    #RMWC,(R2)+
1520    024256  112722  000004                  MOVB    #RMBA,(R2)+
1521    024262  112722  000000                  MOVB    #RMCS1,(R2)+
1522    024266  112722  000200                  MOVB    #200,(R2)+
1523
1524    024272  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        024276  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        024300  000240                          NOP                     ;RETURN HERE IF ERROR
        024302  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        024304  000137  024416                  JMP     190$            ;GO TO 190$ IF ERROR
1525    024310                          130$:
1526
1527                                    ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1528    024310  004737  037024                  JSR     PC,GETSTS
1529    024314  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
1530
1531    024320  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        024324  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        024326  000240                          NOP                     ;RETURN HERE IF ERROR
        024330  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        024332  000137  024416                  JMP     190$            ;GO TO 190$ IF ERROR
1532    024336                          140$:
1533
1534                                    ;VERIFY THE RESULTS OF READ OPERATION
1535    024336  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        024342  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        024344  000240                          NOP                     ;RETURN HERE IF ERROR
        024346  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        024350  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024352  000137  024416                  JMP     190$            ;GO TO 190$ IF ERROR
1536    024356                          150$:
1537    024356  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        024362  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        024364  000240                          NOP                     ;RETURN HERE IF ERROR
        024366  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        024370  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024372  000137  024416                  JMP     190$            ;GO TO 190$ IF ERROR
1538    024376                          160$:
1539    024376  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        024402  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
        024404  000240                          NOP                     ;RETURN HERE IF ERROR
        024406  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        024410  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024412  000137  024416                  JMP     190$            ;GO TO 190$ IF ERROR
1540    024416                          170$:
1541
```

```
                                190$:

                                ;;*****************************************************************
                                ;*TEST 24       READ INVALID SECTOR ADDRESS
                                ;;* *************************************************************
                                TST24:
024416                                  SCOPE                   ;SCOPE CALL
024416  000004                         NOP                     ;START OF TEST
024420  000240                         MOV     #STACK,SP       ;INITIALIZE STACK POINTER
024422  012706  001100                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
024426  013700  001276                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
024432  013701  001464                 MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
024436  012737  000024  001226
1545
1546 024444  012737  000000  001444   MOV     #0,RMDC0        ;CYLINDER = 0
1547 024452  012737  000036  001416   MOV     #30.,RMDA0      ;TRACK = 0, INVALID SECTOR = 30.
1548 024460  012737  000000  001442   MOV     #0,RMOF0        ;18 BIT FORMAT
1549 024466  012737  101206  001414   MOV     #BUFONE,RMBA0   ;CHANGE BUS ADDRESS
1550 024474  012737  177376  001412   MOV     #-258.,RMWC0    ;2 * 256 WORDS (2'S COMP)
1551 024502  012737  000073  001410   MOV     #RH!GO,RMCS10   ;READ HEADER AND DATA
1552 024510                         10$:
1553
1554                                ;PREPARE DEVICE FOR DATA TRANSFER
1555 024510  004737  033352                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
024514  154130                         .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
024516  000404                         BR      20$             ;GO TO 20$ IF NO ERROR
024520  000240                         NOP                     ;RETURN HERE IF ERROR
024522  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
024524  000137  024760                 JMP     90$             ;GO TO 90$ IF ERROR
1556 024530                         20$:
1557
1558                                ;SETUP AND EXECUTE READ HEADER AND DATA COMMAND
1559 024530  012702  001551                MOV     #PUTINX,R2      ;LOAD REGISTER INDEX TABLE
1560 024534  112722  000006                MOVB    #RMDA,(R2)+
1561 024540  112722  000034                MOVB    #RMDC,(R2)+
1562 024544  112722  000032                MOVB    #RMOF,(R2)+
1563 024550  112722  000002                MOVB    #RMWC,(R2)+
1564 024554  112722  000004                MOVB    #RMBA,(R2)+
1565 024560  112722  000000                MOVB    #RMCS1,(R2)+
1566 024564  112722  000200                MOVB    #200,(R2)+      ;SET TERMINATOR BYTE
1567
1568 024570  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
024574  000404                         BR      30$             ;GO TO 30$ IF NO ERROR
024576  000240                         NOP                     ;RETURN HERE IF ERROR
024600  104000                         EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
024602  000137  024760                 JMP     90$             ;GO TO 90$ IF ERROR
1569 024606                         30$:
1570
1571                                ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1572 024606  004737  037024                JSR     PC,GETSTS
1573 024612  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
```

```
1574
1575 024616  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     024622  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
     024624  000240                          NOP                     ;RETURN HERE IF ERROR
     024626  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     024630  000137  024760                  JMP     90$             ;GO TO 90$ IF ERROR
1576 024634                          40$:
1577
1578                                 ;VERIFY RESULTS OF READ COMMAND
1579 024634  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     024640  000405                          BR      50$             ;GO TO 50$ IF NO ERROR
     024642  000240                          NOP                     ;RETURN HERE IF ERROR
     024644  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     024646  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024650  000137  024760                  JMP     90$             ;GO TO 90$ IF ERROR
1580 024654                          50$:
1581 024654  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024660  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
     024662  000240                          NOP                     ;RETURN HERE IF ERROR
     024664  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024666  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024670  000137  024760                  JMP     90$             ;GO TO 90$ IF ERROR
1582 024674                          60$:
1583 024674  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     024700  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
     024702  000240                          NOP                     ;RETURN HERE IF ERROR
     024704  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     024706  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024710  000137  024760                  JMP     90$             ;GO TO 90$ IF ERROR
1584 024714                          70$:
1585
1586                                 ;INCREMENT ADDRESS AND READ NEXT SECTOR
1587 024714  005237  001416                  INC     RMDAO           ;ADVANCE SECTOR COUNT
1588 024720  022737  000037  001416          CMP     #31.,RMDAO      ;DONE ALL SECTORS??
1589 024726  103012                          BHIS    80$             ;NO !!
1590
1591 024730  032737  010000  001442          BIT     #FMT16,RMOFO    ;TEST 16 BIT MODE YET ?
1592 024736  001010                          BNE     90$             ;YES !!
1593 024740  012737  000037  001416          MOV     #31.,RMDAO      ;TRACK = 0, INVALID SECTOR   31.
1594 024746  012737  010000  001442          MOV     #FMT16,RMOFO    ;SET 16 BI  MODE IN OFFSET AND
1595 024754  000137  024510          80$:    JMP     10$             ;TEST AGAIN.
1596
1597 024760                          90$:
1598
1599                                 ;;*******************************************************
                                     ;*TEST 25       READ INVALID TRACK ADDRESS
                                     ;;*******************************************************
     024760                          TST25:
     024760  000004                          SCOPE                   ;SCOPE CALL
     024762  000240                          NOP                     ;START OF TEST
     024764  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     024770  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     024774  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     025000  012737  000025  001226          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1600
1601 025006  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
1602 025014  013737  001332  001416          MOV     LSTRK,RMDAO     ;LOAD LAST TRACK, SECTOR   0
```

```
1603 025022  105237  001417                    INCB    RMDAO+1          ;INCREMENT TO FIRST INVALID TRACK
1604 025026  012737  010000  001442            MOV     #FMT16,RMOFO     ;16 BIT FORMAT
1605 025034  012737  177376  001412            MOV     #-258.,RMWCO     ;2 * 256 WORDS (2'S COMP)
1606 025042  012737  101206  001414            MOV     #BUFONE,RMBAO    ;DATA BUFFER ADDRESS
1607 025050  012737  000073  001410            MOV     #RH.GO,RMCS1O    ;READ HEADER AND DATA
1608 025056                                10$:
1609
1610                                       ;PREPARE DEVICE FOR DATA TRANSFER
1611 025056  004737  033352                    JSR     PC,TSTPRP        ;PREPARE DEVICE FOR TEST
     025062  154130                            .WORD   154130           ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
     025064  000404                            BR      20$              ;GO TO 20$ IF NO ERROR
     025066  000240                            NOP                      ;RETURN HERE IF ERROR
     025070  104000                            EMT                      ;ERPOR # DEFINFD BY TSTPRP SUBROUTINE
     025072  000137  025302                    JMP     80$              ;GO TO 80$ IF ERROR
1612 025076                                20$:
1613
1614                                       ;SETUP AND EXECUTE READ HEADER AND DATA COMMAND
1615 025076  012702  001551                    MOV     #PUTINX,R2       ;LOAD REGISTER INDEX TABLE
1616 025102  112722  000006                    MOVB    #RMDA,(R2)+
1617 025106  112722  000034                    MOVB    #RMDC,(R2)+
1618 025112  112722  000032                    MOVB    #RMOF,(R2)+
1619 025116  112722  000002                    MOVB    #RMWC,(R2)+
1620 025122  112722  000004                    MOVB    #RMBA,(R2)+
1621 025126  112722  000000                    MOVB    #RMCSI,(R2)+
1622 025132  112722  000200                    MOVB    #200,(R2)+       ;SET TERMINATOR BYTE
1623
1624 025136  004737  037360                    JSR     PC,PUT           ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     025142  000404                            BR      30$              ;GO TO 30$ IF NO ERROR
     025144  000240                            NOP                      ;RETURN HERE IF ERROR
     025146  104000                            EMT                      ;ERROR # DEFINED BY PUT SUBROUTINE
     025150  000137  025302                    JMP     80$              ;GO TO 80$ IF ERROR
1625 025154                                30$:
1626
1627                                       ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1628 025154  004737  037024                    JSR     PC,GETSTS
1629 025160  004737  037722                    JSR     PC,TIMOUT        ;WAIT FOR COMMAND TO COMPLETE
1630
1631 025164  004737  037110                    JSR     PC,GET           ;GO READ REGISTER(S) WITH GET SUBROUTINE
     025170  000404                            BR      40$              ;GO TO 40$ IF NO ERROR
     025172  000240                            NOP                      ;RETURN HERE IF ERROR
     025174  104000                            EMT                      ;ERROR # DEFINED BY GET SUBROUTINE
     025176  000137  025302                    JMP     80$              ;GO TO 80$ IF ERROR
1632 025202                                40$:
1633
1634                                       ;VERIFY RESULTS OF READ COMMAND
1635 025202  004737  040106                    JSR     PC,PRIERR        ;GO CHECK FOR PRIMARY ERRORS
     025206  000405                            BR      50$              ;GO TO 50$ IF NO ERROR
     025210  000240                            NOP                      ;RETURN HERE IF ERROR
     025212  104000                            EMT                      ;ERROR # DEFINED BY PRIERR SUBROUTINE
     025214  004736                            JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
```

```
                025216  000137  025302                  JMP     80$             ;GO TO 80$ IF ERROR
     1636       025222                          50$:
     1637       025222  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                025226  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
                025230  000240                          NOP                     ;RETURN HERE IF ERROR
                025232  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                025234  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                025236  000137  025302                  JMP     80$             ;GO TO 80$ IF ERROR
     1638       025242                          60$:
     1639       025242  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                025246  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
                025250  000240                          NOP                     ;RETURN HERE IF ERROR
                025252  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                025254  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                025256  000137  025302                  JMP     80$             ;GO TO 80$ IF ERROR
     1640       025262                          70$:
     1641
     1642
     1643       025262  105237  001417                  INCB    RMDA0+1         ;ADVANCE TRACK COUNT
     1644       025266  123727  001417  000200         CMPB    RMDA0+1,#128.   ;DONE ALL TRACKS??
     1645       025274  101002                          BHI     80$             ;YES!!
     1646       025276  000137  025056                  JMP     10$             ;GO DO NEXT TRACK
     1647       025302                          80$:
     1648
     1649                                       ;;*****************************************************************
                                                ;*TEST 26       READ INVALID CYLINDER ADDRESS
                                                ;;*****************************************************************
                025302                          TST26:
                025302  000004                          SCOPE                   ;SCOPE CALL
                025304  000240                          NOP                     ;START OF TEST
                025306  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                025312  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                025316  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                025322  012737  000026  001226          MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     1650
     1651       025330  012737  001467  001444          MOV     #823.,RMDC0     ;START AT FIRST INVALID CYLINDER
     1652       025336  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
     1653       025344  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
     1654       025352  012737  177376  001412          MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
     1655       025360  012737  101206  001414          MOV     #BUF0NE,RMBA0   ;DATA BUFFER ADDRESS
     1656       025366  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER AND DATA
     1657       025374                          10$:
     1658
     1659
     1660       025374  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                025400  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                025402  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
                025404  000240                          NOP                     ;RETURN HERE IF ERROR
                025406  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                025410  000137  025620                  JMP     80$             ;GO TO 80$ IF ERROR
```

```
1661 025414                          20$:
1662
1663                                 ;SETUP AND EXECUTE READ HEADER AND DATA COMMAND
1664 025414  012702  001551                  MOV     #PUTINX,R2      ;LOAD REGISTER INDEX TABLE
1665 025420  112722  000006                  MOVB    #RMDA,(R2)+
1666 025424  112722  000034                  MOVB    #RMDC,(R2)+
1667 025430  112722  000032                  MOVB    #RMOF,(R2)+
1668 025434  112722  000002                  MOVB    #RMWC,(R2)+
1669 025440  112722  000004                  MOVB    #RMBA,(R2)+
1670 025444  112722  000000                  MOVB    #RMCS1,(R2)+
1671 025450  112722  000200                  MOVB    #200,(R2)+      ;SET TERMINATOR BYTE
1672
1673 025454  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     025460  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
     025462  000240                          NOP                     ;RETURN HERE IF ERROR
     025464  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     025466  000137  025620                  JMP     80$             ;GO TO 80$ IF ERROR
1674 025472                          30$:
1675
1676                                 ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1677 025472  004737  037024                  JSR     PC,GETSTS
1678 025476  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1679
1680 025502  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     025506  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
     025510  000240                          NOP                     ;RETURN HERE IF ERROR
     025512  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     025514  000137  025620                  JMP     80$             ;GO TO 80$ IF ERROR
1681 025520                          40$:
1682
1683                                 ;VERIFY RESULTS OF READ COMMAND
1684 025520  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     025524  000405                          BR      50$             ;GO TO 50$ IF NO ERROR
     025526  000240                          NOP                     ;RETURN HERE IF ERROR
     025530  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     025532  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025534  000137  025620                  JMP     80$             ;GO TO 80$ IF ERROR
1685 025540                          50$:
1686 025540  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     025544  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
     025546  000240                          NOP                     ;RETURN HERE IF ERROR
     025550  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     025552  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025554  000137  025620                  JMP     80$             ;GO TO 80$ IF ERROR
1687 025560                          60$:
1688 025560  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     025564  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
     025566  000240                          NOP                     ;RETURN HERE IF ERROR
     025570  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     025572  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025574  000137  025620                  JMP     80$             ;GO TO 80$ IF ERROR
1689 025600                          70$
1690
1691                                 ;INCREMENT ADDRESS AND READ NEXT CYLINDER
1692 025600  005237  001444                  INC     RMDCO           ;ADVANCE CYLINDER COUNT
1693 025604  023727  001444  002000          CMP     RMDCO,#1024.    ;DONE ALL CYLINDERS??
1694 025612  002002                          BGE     80$             ;YES!!
```

```
1695 025614  000137  025374                    JMP     10$             ;GO DO NEXT SECTOR
1696 025620                            80$:
1697
1698                                   ;;*****************************************************************
                                       ;*TEST 27       FORMAT AT OFFSET
                                       ;;*****************************************************************
     025620                            TST27:
     025620  000004                            SCOPE                   ;SCOPE CALL
     025622  000240                            NOP                     ;START OF TEST
     025624  012706  001100                    MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     025630  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     025634  013701  001464                    MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     025640  012737  000027  001226            MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1699
1700                                   ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
1701 025646  012737  000000  001444            MOV     #0,RMDC0        ;CYLINDER = 0
1702 025654  012737  000000  001416            MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
1703 025662  012737  010000  001442            MOV     #FMT16,RMOF0    ;16 BIT FORMAT
1704 025670  012737  177376  001412            MOV     #-258.,RMWC0    ;2 + 256 WORDS (2'S COMP)
1705 025676  012737  101206  001414            MOV     #BUF ONE,RMBA0  ;DATA BUFFER ADDRESS
1706 025704  012737  000063  001410            MOV     #WH GO,RMCS10   ;WRITE HEADER AND DATA
1707
1708                                   ;VERIFY THAT SECTOR IS NOT BAD
     025712  004737  034276                    JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     025716  000405                            BR      10$             ;GO TO 10$ IF NO ERROR
     025720  104401  063120                    TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     025724  104000                            EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     025726  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1709 025732                            10$:
1710 025732  012737  001416  001174            MOV     #RMDA0,$TMP0    ;USE SECTOR FOR DATA PATTERN
1711 025740  012737  000001  001176            MOV     #1,$TMP1
1712 025746  004737  036224                    JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
1713 025752                            20$:
1714
1715                                   ;PREPARE DEVICE FOR DATA TRANSFER
1716 025752  004737  033352                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     025756  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
     025760  000404                            BR      30$             ;GO TO 30$ IF NO ERROR
     025762  000240                            NOP                     ;RETURN HERE IF ERROR
     025764  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     025766  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1717 025772                            30$:
1718
1719                                   ;OFFSET DEVICE FOR WRITE
1720 025772  012737  000015  001410            MOV     #OFFSET!GO,RMCS10       ;CHANGE TO OFFSET COMMAND
1721 026000  012702  001551                    MOV     #PUTINX,R2      ;WRITE PUT INDEX TABLE
1722 026004  112722  000006                    MOVB    #RMDA,(R2)+
1723 026010  112722  000034                    MOVB    #RMDC,(R2)+
1724 026014  112722  000032                    MOVB    #RMOF,(R2)+
1725 026020  112722  000000                    MOVB    #RMCS1,(R2)+
```

```
1726 026024  112712  000200                    MOVB    #200,(R2)       ;TERMINATE TABLE
1727
1728 026030  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     026034  000404                            BR      35$             ;GO TO 35$ IF NO ERROR
     026036  000240                            NOP                     ;RETURN HERE IF ERROR
     026040  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     026042  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1729 026046                            35$:
1730
1731                                   ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND AT OFFSET
1732 026046  012737  000063  001410            MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA COMMAND
1733 026054  012702  001551                    MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1734 026060  112722  000002                    MOVB    #RMWC,(R2)+
1735 026064  112722  000004                    MOVB    #RMBA,(R2)+
1736 026070  112722  000000                    MOVB    #RMCS1,(R2)+
1737 026074  112722  000200                    MOVB    #200,(R2)+
1738
1739 026100  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     026104  000404                            BR      80$             ;GO TO 80$ IF NO ERROR
     026106  000240                            NOP                     ;RETURN HERE IF ERROR
     026110  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     026112  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1740 026116                            80$:
1741
1742                                   ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1743 026116  004737  037024                    JSR     PC,GETSTS
1744 026122  004737  037722                    JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1745
1746 026126  004737  037110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     026132  000404                            BR      90$             ;GO TO 90$ IF NO ERROR
     026134  000240                            NOP                     ;RETURN HERE IF ERROR
     026136  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     026140  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1747 026144                            90$:
1748
1749                                   ;VERIFY RESULTS OF WRITE COMMAND
1750 026144  004737  040106                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     026150  000405                            BR      100$            ;GO TO 100$ IF NO ERROR
     026152  000240                            NOP                     ;RETURN HERE IF ERROR
     026154  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     026156  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026160  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1751 026164                            100$:
1752 026164  004737  052622                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     026170  000405                            BR      110$            ;GO TO 110$ IF NO ERROR
     026172  000240                            NOP                     ;RETURN HERE IF ERROR
     026174  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     026176  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026200  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1753 026204                            110$:
1754 026204  004737  040740                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     026210  000405                            BR      120$            ;GO TO 120$ IF NO ERROR
     026212  000240                            NOP                     ;RETURN HERE IF ERROR
     026214  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     026216  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026220  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1755 026224                            120$:
```

```
1756                                    ;OFFSET DEVICE FOR READ
1757 026224  012737  000015  001410            MOV     #OFFSET!GO,RMCS10       ;CHANGE TO OFFSET COMMAND
1758 026232  012702  001551                    MOV     #PUTINX,R2      ;WRITE PUT INDEX TABLE
1759 026236  112722  000006                    MOVB    #RMDA,(R2)+
1760 026242  112722  000034                    MOVB    #RMDC,(R2)+
1761 026246  112722  000032                    MOVB    #RMOF,(R2)+
1762 026252  112722  000000                    MOVB    #RMCS1,(R2)+
1763 026256  112722  000200                    MOVB    #200,(R2)+
1764
1765 026262  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     026266  000404                            BR      125$            ;GO TO 125$ IF NO ERROR
     026270  000240                            NOP                     ;RETURN HERE IF ERROR
     026272  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     026274  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1766 026300                            125$:
1767
1768
1769                                    ;READ HEADER AND DATA AT OFFSET
1770 026300  012737  000073  001410            MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
1771 026306  012737  102212  001414            MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
1772 026314  012702  001551                    MOV     #PUTINX,R2      ;WRITE PUT INDEX TABLE
1773 026320  112722  000002                    MOVB    #RMWC,(R2)+
1774 026324  112722  000004                    MOVB    #RMBA,(R2)+
1775 026330  112722  000000                    MOVB    #RMCS1,(R2)+
1776 026334  112722  000200                    MOVB    #200,(R2)+
1777
1778 026340  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     026344  000404                            BR      130$            ;GO TO 130$ IF NO ERROR
     026346  000240                            NOP                     ;RETURN HERE IF ERROR
     026350  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     026352  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1779 026356                            130$:
1780
1781                                    ;WAIT FOR READ TO COMPLETE AND GET STATUS
1782 026356  004737  037722                    JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
1783
1784 026362  004737  037110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     026366  000404                            BR      140$            ;GO TO 140$ IF NO ERROR
     026370  000240                            NOP                     ;RETURN HERE IF ERROR
     026372  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     026374  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1785 026400                            140$:
1786
1787                                    ;VERIFY THE RESULTS OF READ OPERATION
1788 026400  004737  040106                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     026404  000405                            BR      150$            ;GO TO 150$ IF NO ERROR
     026406  000240                            NOP                     ;RETURN HERE IF ERROR
     026410  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     026412  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026414  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
1789 026420                            150$:
1790 026420  004737  052622                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     026424  000405                            BR      160$            ;GO TO 160$ IF NO ERROR
     026426  000240                            NOP                     ;RETURN HERE IF ERROR
     026430  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     026432  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026434  000137  026532                    JMP     190$            ;GO TO 190$ IF ERROR
```

```
026440                          160$:
026440  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
026444  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
026446  000240                          NOP                     ;RETURN HERE IF ERROR
026450  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
026452  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
026454  000137  026532                  JMP     190$            ;GO TO 190$ IF ERROR
026460                          170$:
                                ;VERIFY DATA
026460  004737  036462                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
026464  101206                          .WORD   BUFONE          ;STARTING ADDRESS OF WRITE BUFFER
026466  102212                          .WORD   BUFTWO          ;STARTING ADDRESS OF READ BUFFER
026470  000404                          BR      180$            ;GO TO 180$ IF NO ERROR
026472  000240                          NOP                     ;RETURN HERE IF ERROR
026474  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
026476  000137  026532                  JMP     190$            ;GO TO 190$ IF ERROR
026502                          180$:
026502  032737  000200  001442          BIT     #OFD,RMOFO      ;DONE BOTH OFFSETS??
026510  001010                          BNE     190$            ;YES!!
026512  052737  000200  001442          BIS     #OFD,RMOFO      ;SET FORWARD OFFSET
026520  012737  101206  001414          MOV     #BUFONE,RMBAO   ;CHANGE DATA BUFFER
026526  000137  025752                  JMP     20$
026532                          190$:

                                ;;*******************************************************************
                                ;*TEST 30        IVC FORMAT TEST
                                ;;*******************************************************************
026532                          TST30:
026532  000004                  SCOPE                           ;SCOPE CALL
026534  000240                  NOP                             ;START OF TEST
026536  012706  001100          MOV     #STACK,SP               ;INITIALIZE STACK POINTER
026542  013700  001276          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
026546  013701  001464          MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
026552  012737  000030  001226  MOV     #30,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

                                ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
026560  012737  000000  001444  MOV     #0,RMDCO                ;CYLINDER = 0
026566  012737  000000  001416  MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
026574  012737  010000  001442  MOV     #FMT16,RMOFO            ;16 BIT FORMAT
026602  012737  177376  001412  MOV     #-258.,RMWCO            ;2 + 256 WORDS (2'S COMP)
026610  012737  101206  001414  MOV     #BUFONE,RMBAO           ;DATA BUFFER ADDRESS
026616  012737  000062  001410  MOV     #WH,RMCS1O              ;WRITE HEADER AND DATA

                                ;VERIFY THAT SECTOR IS NOT BAD
026624  004737  034276          JSR     PC,BADSCT               ;CALL BAD SECTOR MODULE
026630  000405                  BR      10$                     ;GO TO 10$ IF NO ERROR
026632  104401  063120          TYPE    ,SCTMSG                 ;TYPE BAD SECTOR MESSAGE
026636  104000                  EMT                             ;ERROR # DEFINED BY BADSCT SUBROUTINE
026640  000137  027354          JMP     190$                    ;GO TO 190$ IF ERROR
026644                          10$:
026644  012737  001416  001174  MOV     #RMDAO,$TMP0            ;USE SECTOR FOR DATA PATTERN
026652  012737  000001  001176  MOV     #1,$TMP1
026660  004737  036224          JSR     PC,GENBUF               ;GO GENERATE DATA BUFFER
026664                          20$:

                                ;PREPARE DEVICE FOR DATA TRANSFER
```

```
1823 026664  004737  033352                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     026670  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
     026672  000404                        BR      30$             ;GO TO 30$ IF NO ERROR
     026674  000240                        NOP                     ;RETURN HERE IF ERROR
     026676  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     026700  000137  027354                JMP     190$            ;GO TO 190$ IF ERROR
1824 026704                         30$:
1825
1826                                ;RESET VOLUME VALID BY SETTING AND RESETTING DIAGNOSTIC MODE
1827 026704  012737  000001  001434        MOV     #DMD,RMMR1O     ;SET DIAGNOSTIC MODE
1828 026712  112737  000024  001551        MOVB    #RMMR1,PUTINX   ;WRITE REGISTER INDEX TABLE
1829 026720  112737  000200  001552        MOVB    #200,PUTINX+1
1830
1831 026726  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     026732  000404                        BR      35$             ;GO TO 35$ IF NO ERROR
     026734  000240                        NOP                     ;RETURN HERE IF ERROR
     026736  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     026740  000137  027354                JMP     190$            ;GO TO 190$ IF ERROR
1832 026744                         35$:
1833
1834                                ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
1835 026744  012737  000063  001410        MOV     #WH!GO,RMCS1O   ;WRITE HEADER AND DATA
1836 026752  012737  000000  001434        MOV     #0,RMMR1O       ;RESET DIAGNOSTIC MODE
1837 026760  012702  001551                MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
1838 026764  112722  000024                MOVB    #RMMR1,(R2)+
1839 026770  112722  000006                MOVB    #RMDA,(R2)+
1840 026774  112722  000034                MOVB    #RMDC,(R2)+
1841 027000  112722  000032                MOVB    #RMOF,(R2)+
1842 027004  112722  000002                MOVB    #RMWC,(R2)+
1843 027010  112722  000004                MOVB    #RMBA,(R2)+
1844 027014  112722  000000                MOVB    #RMCS1,(R2)+
1845 027020  112722  000200                MOVB    #200,(R2)+
1846
1847 027024  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027030  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
     027032  000240                        NOP                     ;RETURN HERE IF ERROR
     027034  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     027036  000137  027354                JMP     190$            ;GO TO 190$ IF ERROR
1848 027042                         80$:
1849
1850                                ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1851 027042  004737  037024                JSR     PC,GETSTS
1852 027046  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
1853
1854 027052  004737  037110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027056  000404                        BR      90$             ;GO TO 90$ IF NO ERROR
     027060  000240                        NOP                     ;RETURN HERE IF ERROR
     027062  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027064  000137  027354                JMP     190$            ;GO TO 190$ IF ERROR
1855 027070                         90$:
```

```
1856
1857                                    ;VERIFY RESULTS OF WRITE COMMAND
1858 027070  004737  040106                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     027074  000405                              BR      100$            ;GO TO 100$ IF NO ERROR
     027076  000240                              NOP                     ;RETURN HERE IF ERROR
     027100  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     027102  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027104  000137  027354                      JMP     190$            ;GO TO 190$ IF ERROR
1859 027110                             100$:
1860 027110  032737  010000  001376              BIT     #IVC,RMER2I     ;IS IVC STATUS SET??
1861 027116  001012                              BNE     110$            ;YES!!
1862 027120  013737  001376  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS FOR TYPEOUT
1863 027126  013737  001376  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
1864 027134  052737  010000  001140              BIS     #IVC,$GDDAT
1865 027142  104342                              EMT     342
1866 027144                             110$:
1867 027144  004737  040740                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     027150  000405                              BR      120$            ;GO TO 120$ IF NO ERROR
     027152  000240                              NOP                     ;RETURN HERE IF ERROR
     027154  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     027156  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027160  000137  027354                      JMP     190$            ;GO TO 190$ IF ERROR
1868 027164                             120$:
1869
1870                                    ;CLEAR IVC ERROR
1871 027164  004737  033352                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     027170  040100                              .WORD   040100          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                         ;CLEAR CONTROLLER & SELECT DEVICE
                                                                         ;VERIFY CONTROLLER CLEAR OPERATION
     027172  000404                              BR      125$            ;GO TO 125$ IF NO ERROR
     027174  000240                              NOP                     ;RETURN HERE IF ERROR
     027176  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     027200  000137  027354                      JMP     190$            ;GO TO 190$ IF ERROR
1872 027204                             125$:
1873
1874                                    ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
1875 027204  012737  000073  001410              MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
1876 027212  012737  102212  001414              MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
1877
1878 027220  004737  037360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027224  000404                              BR      130$            ;GO TO 130$ IF NO ERROR
     027226  000240                              NOP                     ;RETURN HERE IF ERROR
     027230  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     027232  000137  027354                      JMP     190$            ;GO TO 190$ IF ERROR
1879 027236                             130$:
1880
1881                                    ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
1882 027236  004737  037024                      JSR     PC,GETSTS
1883 027242  004737  037722                      JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
1884
1885 027246  004737  037110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027252  000404                              BR      140$            ;GO TO 140$ IF NO ERROR
     027254  000240                              NOP                     ;RETURN HERE IF ERROR
     027256  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027260  000137  027354                      JMP     190$            ;GO TO 190$ IF ERROR
1886 027264                             140$:
1887
```

```
                                        ;VERIFY THE RESULTS OF READ OPERATION
1889 027264  004737  040106             JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     027270  000405                     BR      150$            ;GO TO 150$ IF NO ERROR
     027272  000240                     NOP                     ;RETURN HERE IF ERROR
     027274  104000                     EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     027276  004736                     JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027300  000137  027354             JMP     190$            ;GO TO 190$ IF ERROR
1890 027304                             150$:
1891 027304  032737  010000  001376     BIT     #IVC,RMER2I     ;IS IVC STATUS SET??
1892 027312  001012                     BNE     160$            ;YES!!
1893 027314  013737  001376  001142     MOV     RMER2I,$BDDAT   ;RECEIVED STATUS FOR TYPEOUT
1894 027322  013737  001376  001140     MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
1895 027330  052737  010000  001140     BIS     #IVC,$GDDAT
1896 027336  104342                     EMT     342
1897 027340                             160$:
1898 027340  004737  040740             JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     027344  000403                     BR      170$            ;GO TO 170$ IF NO ERROR
     027346  000240                     NOP                     ;RETURN HERE IF ERROR
     027350  104000                     EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     027352  004736                     JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1899 027354                             170$:
1900
1901 027354                             190$:
1902
1903                                    ;;*******************************************************************
                                        ;*TEST 31       FORMAT ERROR TEST
                                        ;;*******************************************************************
     027354                             TST31:
     027354  000004                     SCOPE                   ;SCOPE CALL
     027356  000240                     NOP                     ;START OF TEST
     027360  012706  001100             MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     027364  013700  001276             MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     027370  013701  001464             MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     027374  012737  000031  001226     MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1904
1905                                    ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
1906 027402  012737  000000  001444     MOV     #0,RMDCO        ;CYLINDER = 0
1907 027410  012737  000000  001416     MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
1908 027416  012737  000000  001442     MOV     #0,RMOFO        ;18 BIT FORMAT
1909 027424  012737  177376  001412     MOV     #-258.,RMWCO    ;2 + 256 WORDS (2'S COMP)
1910 027432                             5$:
1911 027432  012737  101206  001414     MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
1912 027440  012737  000062  001410     MOV     #WH,RMCS1O      ;WRITE HEADER AND DATA
1913
1914                                    ;VERIFY THAT SECTOR IS NOT BAD
     027446  004737  034276             JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     027452  000405                     BR      10$             ;GO TO 10$ IF NO ERROR
     027454  104401  063120             TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     027460  104000                     EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     027462  000137  030136             JMP     180$            ;GO TO 180$ IF ERROR
1915 027466                             10$:
1916 027466  012737  064620  001174     MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
1917 027474  012737  000001  001176     MOV     #1,$TMP1
1918 027502  004737  036224             JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
1919 027506                             20$:
1920
1921                                    ;PREPARE DEVICE FOR DATA TRANSFER
```

```
 1922 027506  004737  033352                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      027512  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
      027514  000404                        BR      30$             ;GO TO 30$ IF NO ERROR
      027516  000240                        NOP                     ;RETURN HERE IF ERROR
      027520  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      027522  000137  030136                JMP     180$            ;GO TO 180$ IF ERROR
 1923 027526                        30$:
 1924
 1925                               ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
 1926 027526  012737  000063  001410        MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
 1927 027534  012702  001551                MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
 1928 027540  112722  000006                MOVB    #RMDA,(R2)+
 1929 027544  112722  000034                MOVB    #RMDC,(R2)+
 1930 027550  112722  000032                MOVB    #RMOF,(R2)+
 1931 027554  112722  000002                MOVB    #RMWC,(R2)+
 1932 027560  112722  000004                MOVB    #RMBA,(R2)+
 1933 027564  112722  000000                MOVB    #RMCS1,(R2)+
 1934 027570  112722  000200                MOVB    #200,(R2)+
 1935
 1936 027574  004737  037360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      027600  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
      027602  000240                        NOP                     ;RETURN HERE IF ERROR
      027604  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      027606  000137  030136                JMP     180$            ;GO TO 180$ IF ERROR
 1937 027612                        80$:
 1938
 1939                               ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
 1940 027612  004737  037024                JSR     PC,GETSTS
 1941 027616  004737  037722                JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
 1942
 1943 027622  004737  037110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      027626  000404                        BR      90$             ;GO TO 90$ IF NO ERROR
      027630  000240                        NOP                     ;RETURN HERE IF ERROR
      027632  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      027634  000137  030136                JMP     180$            ;GO TO 180$ IF ERROR
 1944 027640                        90$:
 1945
 1946                               ;VERIFY RESULTS OF WRITE COMMAND
 1947 027640  004737  040106                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      027644  000405                        BR      100$            ;GO TO 100$ IF NO ERROR
      027646  000240                        NOP                     ;RETURN HERE IF ERROR
      027650  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      027652  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      027654  000137  030136                JMP     180$            ;GO TO 180$ IF ERROR
 1948 027660                        100$:
 1949 027660  004737  052622                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      027664  000405                        BR      110$            ;GO TO 110$ IF NO ERROR
      027666  000240                        NOP                     ;RETURN HERE IF ERROR
      027670  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      027672  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
                027674  000137  030136                  JMP     180$            ;GO TO 180$ IF ERROR
1950            027700                          110$:
1951            027700  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                027704  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
                027706  000240                          NOP                     ;RETURN HERE IF ERROR
                027710  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                027712  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                027714  000137  030136                  JMP     180$            ;GO TO 180$ IF ERROR
1952            027720                          120$:
1953
1954                                            ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN IN OPPOSITE FORMAT
1955                                            ;TO CHECK THAT "FER" SETS
1956            027720  005137  001442                  COM     RMOFO           ;CHANGE TO OPPOSITE FORMAT
1957            027724  042737  167777  001442          BIC     #^C<FMT16>,RMOFO
1958            027732  012737  102212  001414          MOV     #BUFTWO,RMBAO   ;CHANGE BUS ADDRESS
1959            027740  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
1960
1961            027746  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                027752  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
                027754  000240                          NOP                     ;RETURN HERE IF ERROR
                027756  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                027760  000137  030136                  JMP     180$            ;GO TO 180$ IF ERROR
1962            027764                          130$:
1963
1964                                            ;WAIT FOR READ TO COMPLETE AND GET STATUS
1965            027764  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
1966
1967            027770  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                027774  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
                027776  000240                          NOP                     ;RETURN HERE IF ERROR
                030000  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                030002  000137  030136                  JMP     180$            ;GO TO 180$ IF ERROR
1968            030006                          140$:
1969
1970                                            ;VERIFY THE RESULTS OF READ OPERATION
1971            030006  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                030012  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                030014  000240                          NOP                     ;RETURN HERE IF ERROR
                030016  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                030020  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                030022  000137  030136                  JMP     180$            ;GO TO 180$ IF ERROR
1972            030026                          150$:
1973
1974                                            ;VERIFY THAT FORMAT ERROR IS SET
1975            030026  032737  000020  001350          BIT     #FER,RMER11     ;IS FORMAT ERROR SET??
1976            030034  001022                          BNE     160$            ;YES!!
1977
1978            030036  004737  05<622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                030042  000405                          BR      155$            ;GO TO 155$ IF NO ERROR
                030044  000240                          NOP                     ;RETURN HERE IF ERROR
                030046  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                030050  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                030052  000137  030136                  JMP     180$            ;GO TO 180$ IF ERROR
1979
1980            030056  013737  001350  001142  155$:   MOV     RMER11,$BDDAT   ;BAD DATA FOR TYPEOUT
1981            030064  013737  001350  001140          MOV     RMER11,$GDDAT   ;EXPECTED DATA
1982            030072  052737  000020  001140          BIS     #FER,$GDDAT
```

```
1983 030100  104343                        EMT     343
1984 030102                          160$:
1985 030102  004737  040740                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     030106  000405                        BR      170$            ;GO TO 170$ IF NO ERROR
     030110  000240                        NOP                     ;RETURN HERE IF ERROR
     030112  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     030114  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     030116  000137  030136                JMP     180$            ;GO TO 180$ IF ERROR
1986 030122                          170$:
1987 030122  032737  010000  001442       BIT     #FMT16,RMOFO    ;TEST WRITE 16 BIT FORMAT AND
1988                                                              ;READ 18 BIT MODE ?
1989 030130  001402                        BEQ     180$            ;BR IF YES
1990 030132  000137  027432                JMP     5$              ;TEST AGAIN
1991 030136                          180$:
1992
1993                                 ;;*****************************************************************
                                     ;*TEST 32       FORMAT HCE, FIRST HEADER WORD
                                     ;;*****************************************************************
     030136                          TST32:
     030136  000004                        SCOPE                   ;SCOPE CALL
     030140  000240                        NOP                     ;START OF TEST
     030142  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     030146  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     030152  013701  001464                MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     030156  012737  000032  001226        MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1994
1995 030164  012737  000001  031134        MOV     #1,230$         ;INIT BIT POSITION
1996
1997                                 ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
1998 030172                          10$:
1999 030172  012737  010000  001442        MOV     #FMT16,RMOFO    ;16 BIT FORMAT
2000 030200  012737  000000  001444        MOV     #0,RMDCO        ;CYLINDER = 0
2001 030206  012737  000000  001416        MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
2002 030214  012737  177376  001412        MOV     #-258.,RMWCO    ;2 + 256 WORDS (2'S COMP)
2003 030222  012737  101206  001414        MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
2004 030230  012737  000062  001410        MOV     #WH,RMCS1O      ;WRITE HEADER AND DATA
2005
2006                                 ;VERIFY THAT SECTOR IS NOT BAD
     030236  004737  034276                JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     030242  000405                        BR      20$             ;GO TO 20$ IF NO ERROR
     030244  104401  063120                TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     030250  104000                        EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     030252  000137  031306                JMP     270$            ;GO TO 270$ IF ERROR
2007 030256                          20$:
2008 030256  C12737  064620  001174        MOV     #ZEROS,$TMP0    ;USE ALL ZEROS DATA PATTERN
2009 030264  012737  000001  001176        MOV     #1,$TMP1
2010 030272  004737  036224                JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
2011 030276                          30$:
2012
2013                                 ;PREPARE DEVICE FOR DATA TRANSFER
2014 030276  004737  033352                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     030302  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
     030304  000404                     BR      40$            ;GO TO 40$ IF NO ERROR
     030306  000240                     NOP                    ;RETURN HERE IF ERROR
     030310  104000                     EMT                    ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     030312  000137  031306             JMP     270$           ;GO TO 270$ IF ERROR
2015 030316                     40$.
2016
2017                            ;COMPLEMENT DATA BIT IN FIRST HEADER WORD
2018 030316  033737  031134  101206     BIT     230$,BUFONE    ;IS BIT IN HEADER ON??
2019 030324  001404                     BEQ     50$            ;NO!!
2020 030326  043737  031134  101206     BIC     230$,BUFONE    ;RESET BIT IN HEADER
2021 030334  000403                     BR      60$
2022 030336  053737  031134  101206 50$: BIS     230$,BUFONE    ;SET BIT IN HEADER
2023
2024                            ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
2025 030344                     60$:
2026 030344  012737  000063  001410     MOV     #WH!GO,RMCS10  ;WRITE HEADER AND DATA
2027 030352  012702  001551             MOV     #PUTINX,R2     ;WRITE REGISTER INDEX TABLE
2028 030356  112722  000006             MOVB    #RMDA,(R2)+
2029 030362  112722  000034             MOVB    #RMDC,(R2)+
2030 030366  112722  000032             MOVB    #RMOF,(R2)+
2031 030372  112722  000002             MOVB    #RMWC,(R2)+
2032 030376  112722  000004             MOVB    #RMBA,(R2)+
2033 030402  112722  000000             MOVB    #RMCS1,(R2)+
2034 030406  112722  000200             MOVB    #200,(R2)+
2035
2036 030412  004737  037360             JSR     PC,PUT         ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     030416  000404                     BR      70$            ;GO TO 70$ IF NO ERROR
     030420  000240                     NOP                    ;RETURN HERE IF ERROR
     030422  104000                     EMT                    ;ERROR # DEFINED BY PUT SUBROUTINE
     030424  000137  031132             JMP     220$           ;GO TO 220$ IF ERROR
2037 030430                     70$:
2038
2039                            ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
2040 030430  004737  037024             JSR     PC,GETSTS
2041 030434  004737  037722             JSR     PC,TIMOUT      ;WAIT FOR COMMAND TO COMPLETE
2042
2043 030440  004737  037110             JSR     PC,GET         ;GO READ REGISTER(S) WITH GET SUBROUTINE
     030444  000404                     BR      80$            ;GO TO 80$ IF NO ERROR
     030446  000240                     NOP                    ;RETURN HERE IF ERROR
     030450  104000                     EMT                    ;ERROR # DEFINED BY GET SUBROUTINE
     030452  000137  031132             JMP     220$           ;GO TO 220$ IF ERROR
2044 030456                     80$:
2045
2046                            ;VERIFY RESULTS OF WRITE COMMAND
2047 030456  004737  040106             JSR     PC,PRIERR      ;GO CHECK FOR PRIMARY ERRORS
     030462  000405                     BR      90$            ;GO TO 90$ IF NO ERROR
     030464  000240                     NOP                    ;RETURN HERE IF ERROR
     030466  104000                     EMT                    ;ERROR # DEFINED BY PRIERR SUBROUTINE
     030470  004736                     JSR     PC,@(SP)+      ;GO BACK FOR MORE ERROR CHECKS
     030472  000137  031132             JMP     220$           ;GO TO 220$ IF ERROR
2048 030476                     90$:
2049 030476  004737  052622             JSR     PC,DTASTS      ;GO VERIFY RESULTS OF DATA TRANSFER
     030502  000405                     BR      100$           ;GO TO 100$ IF NO ERROR
     030504  000240                     NOP                    ;RETURN HERE IF ERROR
     030506  104000                     EMT                    ;ERROR # DEFINED BY DTASTS SUBROUTINE
```

```
        030510  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030512  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
  2050  030516                          100$:
  2051  030516  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        030522  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        030524  000240                          NOP                     ;RETURN HERE IF ERROR
        030526  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        030530  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030532  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
  2052  030536                          110$:
  2053
  2054                                  ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
  2055  030536  012737  000073  001410          MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
  2056  030544  012737  102212  001414          MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
  2057
  2058  030552  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        030556  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
        030560  000240                          NOP                     ;RETURN HERE IF ERROR
        030562  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        030564  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
  2059  030570                          120$:
  2060
  2061                                  ;WAIT FOR READ TO COMPLETE AND GET STATUS
  2062  030570  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
  2063
  2064  030574  004737  037110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        030600  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        030602  000240                          NOP                     ;RETURN HERE IF ERROR
        030604  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        030606  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
  2065  030612                          130$:
  2066
  2067                                  ;VERIFY THE RESULTS OF READ OPERATION
  2068  030612  004737  040106                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        030616  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        030620  000240                          NOP                     ;RETURN HERE IF ERROR
        030622  10400C                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        030624  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030626  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
  2069  030632                          140$:
  2070  030632  032737  140000  031134          BIT     #MSE!USE,230$   ;SHOULD BSE BE SET ?
  2071  030640  001033                          BNE     160$            ;YES !!
  2072  030642  032737  010000  031134          BIT     #FMT16,230$     ;IS THIS FER ??
  2073  030650  001456                          BEQ     180$            ;NO !!
  2074
  2075                                  ;VERIFY THAT FER IS SET
  2076  030652  032737  000020  001350          BIT     #FER,RMER11     ;IS FER SET ??
  2077  030660  001107                          BNE     200$            ;YES !!
  2078
  2079  030662  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        030666  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        030670  000240                          NOP                     ;RETURN HERE IF ERROR
        030672  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        030674  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030676  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
  2080  030702                          150$:
  2081  030702  013737  001350  001142          MOV     RMER11,$BDDAT   ;RECEIVED STATUS
```

```
2082 030710  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
2083 030716  052737  000020  001140          BIS     #FER,$GDDAT
2084 030724  104343                          EMT     343
2085 030726  000501                          BR      220$
2086
2087                                  ;VERIFY THAT BSE IS SET
2088 030730                           160$:
2089 030730  032737  100000  001376          BIT     #BSE,RMER2I     ;IS BSE SET ??
2090 030736  001060                          BNE     200$            ;YES !!
2091
2092 030740  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     030744  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
     030746  000240                          NOP                     ;RETURN HERE IF ERROR
     030750  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     030752  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     030754  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
2093 030760                           170$:
2094 030760  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
2095 030766  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
2096 030774  052737  100000  001140          BIS     #BSE,$GDDAT
2097 031002  104345                          EMT     345
2098 031004  000452                          BR      220$            ;SKIP REST OF TEST
2099
2100                                  ;VERIFY THAT HCE IS SET
2101 031006                           180$:
2102 031006  032737  000200  001350          BIT     #HCE,RMER1I     ;IS "HCE" SET??
2103 031014  001031                          BNE     200$            ;YES!!
2104
2105 031016  004737  052622                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     031022  000405                          BR      190$            ;GO TO 190$ IF NO ERROR
     031024  000240                          NOP                     ;RETURN HERE IF ERROR
     031026  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     031030  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031032  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
2106 031036                           190$:
2107 031036  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS FOR TYPEOUT
2108 031044  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
2109 031052  052737  000200  001140          BIS     #HCE,$GDDAT
2110 031060  012737  000001  001174          MOV     #1,$TMP0        ;GET HEADER WORD NUMBER
2111 031066  013737  031134  001176          MOV     230$,$TMP1      ;GET FAILING BIT POSITION
2112 031074  104344                          EMT     344
2113 031076  000415                          BR      220$
2114 031100                           200$:
2115
2116                                  ;CHECK FOR OTHER ERRORS
2117 031100  004737  040740                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     031104  000405                          BR      210$            ;GO TO 210$ IF NO ERROR
     031106  000240                          NOP                     ;RETURN HERE IF ERROR
     031110  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     031112  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031114  000137  031132                  JMP     220$            ;GO TO 220$ IF ERROR
2118 031120                           210$:
2119
2120                                  ;ADVANCE THE BIT POSITION AND FORMAT AGAIN IF NOT DONE
2121 031120  006337  031134                  ASL     230$            ;SHIFT TO NEXT BIT POSITION
2122 031124  001404                          BEQ     240$            ;EXIT IF DONE
2123 031126  000137  030172                  JMP     10$             ;GO DO NEXT SECTOR
```

```
2124
2125 031132  000465                  220$:   BR      270$            ;JUMP TO NEXT TEST
2126 031134  000000                  230$:   .WORD   0               ;STORAGE FOR BIT POSITION
2127
2128                                  ;;*****************************************************************
2129                                  ;*REFORMAT SECTOR THAT WAS WRITTEN WITH BAD HEADER
2130                                  ;;*****************************************************************
2131 031136                          240$:
2132 031136  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT MODE
2133 031144  012737  177776  001412          MOV     #-2,RMWCO       ;ONLY TWO HEADER WORDS
2134 031152  012737  101206  001414          MOV     #BUFONE,RMBAO   ;SELECT THE BUFFER
2135 031160  012737  000062  001410          MOV     #WH,RMCS1O      ;WRITE HEAD AND DATA COMMAND
2136 031166  004737  036224                  JSR     PC,GENBUF       ;GENERATE THE BUFFER PATTERN
2137
2138 031172  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     031176  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
     031200  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
     031202  000240                          NOP                     ;RETURN HERE IF ERROR
     031204  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     031206  000137  031212                  JMP     250$            ;GO TO 250$ IF ERROR
2139 031212                          250$:
2140 031212  012737  000063  001410          MOV     #WH!GO,RMCS1O   ;SET THE GO BIT
2141 031220  012702  001551                  MOV     #PUTINX,R2      ;PUT THE REGISTER ADDRESS INTO TABLE
2142 031224  112722  000006                  MOVB    #RMDA,(R2)+
2143 031230  112722  000034                  MOVB    #RMDC,(R2)+
2144 031234  112722  000032                  MOVB    #RMOF,(R2)+
2145 031240  112722  000002                  MOVB    #RMWC,(R2)+
2146 031244  112722  000004                  MOVB    #RMBA,(R2)+
2147 031250  112722  000000                  MOVB    #RMCS1,(R2)+
2148 031254  112722  000200                  MOVB    #200,(R2)+      ;TERMINATOR
2149 031260  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031264  000404                          BR      260$            ;GO TO 260$ IF NO ERROR
     031266  000240                          NOP                     ;RETURN HERE IF ERROR
     031270  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     031272  000137  031306                  JMP     270$            ;GO TO 270$ IF ERROR
2150 031276                          260$:
2151 031276  000240                          NOP
2152 031300  004737  037722                  JSR     PC,TIMOUT       ;WAIT UNTIL IT FINISH
2153 031304  000240                          NOP
2154 031306                          270$:
2155
2156                                  ;;*****************************************************************
                                      ;*TEST 33       FORMAT HCE, SECOND HEADER WORD
                                      ;;*****************************************************************
     031306                           TST33:
     031306  000004                          SCOPE                   ;SCOPE CALL
     031310  000240                          NOP                     ;START OF TEST
     031312  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     031316  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     031322  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
```

```
        031326  012737  000033  001226          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  2157
  2158  031334  012737  000001  032130          MOV     #1,190$         ;INIT BIT POSITION
  2159
  2160                                  ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
  2161  031342                          10$:
  2162  031342  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
  2163  031350  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
  2164  031356  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
  2165  031364  012737  177376  001412          MOV     #-258.,RMWCO    ;2 + 256 WORDS (2'S COMP)
  2166  031372  012737  101206  001414          MOV     #BUFONE,RMBAO   ;DATA BUFFER ADDRESS
  2167  031400  012737  000062  001410          MOV     #WH,RMCS1O      ;WRITE HEADER AND DATA
  2168
  2169                                  ;VERIFY THAT SECTOR IS NOT BAD
        031406  004737  034276                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        031412  000405                  BR      20$             ;GO TO 20$ IF NO ERROR
        031414  104401  063120          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        031420  104000                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        031422  000137  032302          JMP     230$            ;GO TO 230$ IF ERROR
  2170  031426                          20$:
  2171  031426  012737  064620  001174          MOV     #ZEROS,$TMPO    ;USE ALL ZEROS DATA PATTERN
  2172  031434  012737  000001  001176          MOV     #1,$TMP1
  2173  031442  004737  036224                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
  2174  031446                          30$:
  2175
  2176                                  ;PREPARE DEVICE FOR DATA TRANSFER
  2177  031446  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        031452  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
        031454  000404                  BR      40$             ;GO TO 40$ IF NO ERROR
        031456  000240                  NOP                     ;RETURN HERE IF ERROR
        031460  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        031462  000137  032302          JMP     230$            ;GO TO 230$ IF ERROR
  2178
  2179                                  ;COMPLEMENT DATA BIT IN SECOND HEADER WORD
  2180  031466                          40$:
  2181  031466  033737  032130  101210          BIT     190$,BUFONE+2   ;IS BIT IN HEADER ON??
  2182  031474  001404                  BEQ     50$             ;NO!!
  2183  031476  043737  032130  101210          BIC     190$,BUFONE+2   ;RESET BIT IN HEADER
  2184  031504  000403                  BR      60$
  2185  031506  053737  032130  101210  50$:    BIS     190$,BUFONE+2   ;SET BIT IN HEADER
  2186
  2187                                  ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
  2188  031514                          60$:
  2189  031514  012737  000063  001410          MOV     #WH!GO,RMCS1O   ;WRITE HEADER AND DATA
  2190  031522  012702  001551                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
  2191  031526  112722  000006                  MOVB    #RMDA,(R2)+
  2192  031532  112722  000034                  MOVB    #RMDC,(R2)+
  2193  031536  112722  000032                  MOVB    #RMOF,(R2)+
  2194  031542  112722  000002                  MOVB    #RMWC,(R2)+
  2195  031546  112722  000004                  MOVB    #RMBA,(R2)+
```

```
2196 031552  112722  000000                    MOVB    #RMCS1,(R2)+
2197 031556  112722  000200                    MOVB    #200,(R2)+
2198
2199 031562  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031566  000404                            BR      70$             ;GO TO 70$ IF NO ERROR
     031570  000240                            NOP                     ;RETURN HERE IF ERROR
     031572  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     031574  000137  032126                    JMP     180$            ;GO TO 180$ IF ERROR
2200 031600                              70$:
2201
2202                                     ;SETUP INPUT REGISTER BUFFER FOR READING STATUS
2203 031600  004737  037024                    JSR     PC,GETSTS
2204 031604  004737  037722                    JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE
2205
2206 031610  004737  037110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     031614  000404                            BR      80$             ;GO TO 80$ IF NO ERROR
     031616  000240                            NOP                     ;RETURN HERE IF ERROR
     031620  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     031622  000137  032126                    JMP     180$            ;GO TO 180$ IF ERROR
2207 031626                              80$:
2208
2209
2210 031626  004737  040106                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     031632  000405                            BR      90$             ;GO TO 90$ IF NO ERROR
     031634  000240                            NOP                     ;RETURN HERE IF ERROR
     031636  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     031640  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031642  000137  032126                    JMP     180$            ;GO TO 180$ IF ERROR
2211 031646                              90$:
2212 031646  004737  052622                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     031652  000405                            BR      100$            ;GO TO 100$ IF NO ERROR
     031654  000240                            NOP                     ;RETURN HERE IF ERROR
     031656  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     031660  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031662  000137  032126                    JMP     180$            ;GO TO 180$ IF ERROR
2213 031666                              100$:
2214 031666  004737  040740                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     031672  000405                            BR      110$            ;GO TO 110$ IF NO ERROR
     031674  000240                            NOP                     ;RETURN HERE IF ERROR
     031676  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     031700  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031702  000137  032126                    JMP     180$            ;GO TO 180$ IF ERROR
2215 031706                              110$:
2216
2217                                     ;READ HEADER AND DATA FOR SECTOR JUST WRITTEN
2218 031706  012737  000073  001410            MOV     #RH!GO,RMCS10   ;READ HEADER & DATA COMMAND
2219 031714  012737  102212  001414            MOV     #BUFTWO,RMBA0   ;CHANGE BUS ADDRESS
2220
2221 031722  004737  037360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031726  000404                            BR      120$            ;GO TO 120$ IF NO ERROR
     031730  000240                            NOP                     ;RETURN HERE IF ERROR
     031732  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     031734  000137  032126                    JMP     180$            ;GO TO 180$ IF ERROR
2222 031740                              120$:
2223
2224                                     ;WAIT FOR READ TO COMPLETE AND GET STATUS
2225 031740  004737  037722                    JSR     PC,TIMOUT       ;WAIT FOR READ TO COMPLETE
```

```
                                                ;GO READ REGISTER(S) WITH GET SUBROUTINE
031744  004737  037110          JSR     PC,GET
031750  000404                  BR      130$            ;GO TO 130$ IF NO ERROR
031752  000240                  NOP                     ;RETURN HERE IF ERROR
031754  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
031756  000137  032126          JMP     180$            ;GO TO 180$ IF ERROR
031762                  130$:
                        ;VERIFY THE RESULTS OF READ OPERATION
031762  004737  040106          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
031766  000405                  BR      140$            ;GO TO 140$ IF NO ERROR
031770  000240                  NOP                     ;RETURN HERE IF ERROR
031772  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
031774  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
031776  000137  032126          JMP     180$            ;GO TO 180$ IF ERROR
032002                  140$:
                        ;VERIFY THAT HEADER COMPARE ERROR IS SET
032002  032737  000200  001350  BIT     #HCE,RMER1I     ;IS "HCE" SET??
032010  001031                  BNE     160$            ;YES!!
                                                ;GO VERIFY RESULTS OF DATA TRANSFER
032012  004737  052622          JSR     PC,DTASTS
032016  000405                  BR      150$            ;GO TO 150$ IF NO ERROR
032020  000240                  NOP                     ;RETURN HERE IF ERROR
032022  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
032024  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
032026  000137  032126          JMP     180$            ;GO TO 180$ IF ERROR
032032                  150$:
032032  013737  001350  001142  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS FOR TYPEOUT
032040  013737  001350  001140  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
032046  052737  000200  001140  BIS     #HCE,$GDDAT
032054  012737  000002  001174  MOV     #2,$TMP0        ;GET HEADER WORD NUMBER
032062  013737  032130  001176  MOV     190$,$TMP1      ;GET FAILING BIT POSITION
032070  104344                  EMT     344
032072  000415                  BR      180$
032074                  160$:

                        ;CHECK FOR OTHER ERRORS
032074  004737  040740          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
032100  000405                  BR      170$            ;GO TO 170$ IF NO ERROR
032102  000240                  NOP                     ;RETURN HERE IF ERROR
032104  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
032106  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
032110  000137  032126          JMP     180$            ;GO TO 180$ IF ERROR
032114                  170$:

                        ;ADVANCE THE BIT POSITION AND FORMAT NEXT SECTOR IF NOT DONE
032114  006337  032130          ASL     190$            ;SHIFT TO NEXT BIT POSITION
032120  001404                  BEQ     200$            ;EXIT IF DONE
032122  000137  031342          JMP     10$             ;GO DO NEXT SECTOR

032126  000465          180$:   BR      230$            ;JUMP OVER STORAGE
032130  000000          190$:   .WORD   0               ;STORAGE FOR BIT POSITION

                        ;;*****************************************************************
                        ;*REFORMAT SECTOR THAT WAS WRITTEN WITH BAD HEADER
                        ;;*****************************************************************
```

```
2264 032132                                  200$:
2265 032132  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT MODE
2266 032140  012737  177776  001412          MOV     #-2,RMWCO       ;2 HEADER WORDS ONLY
2267 032146  012737  101206  001414          MOV     #BUFONE,RMBAO   ;BUFFER ADDRESS
2268 032154  012737  000062  001410          MOV     #WH,RMCS1O      ;WRITE HEAD AND DATA COMMAND
2269 032162  004737  036224                  JSR     PC,GENBUF       ;SET UP THE BUFFER
2270
2271 032166  004737  033352                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     032172  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                     ;VERIFY RECALIBRATION
     032174  000404                          BR      210$            ;GO TO 210$ IF NO ERROR
     032176  000240                          NOP                     ;RETURN HERE IF ERROR
     032200  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     032202  000137  032206                  JMP     210$            ;GO TO 210$ IF ERROR
2272 032206                                  210$:
2273 032206  012737  000063  001410          MOV     #WH!GO,RMCS1O   ;SET GO BIT
2274 032214  012702  001551                  MOV     #PUTINX,R2      ;TABLE ADDRESS
2275 032220  112722  000006                  MOVB    #RMDA,(R2)+
2276 032224  112722  000034                  MOVB    #RMDC,(R2)+
2277 032230  112722  000032                  MOVB    #RMOF,(R2)+
2278 032234  112722  000004                  MOVB    #RMBA,(R2)+
2279 032240  112722  000002                  MOVB    #RMWC,(R2)+
2280 032244  112722  000000                  MOVB    #RMCS1,(R2)+
2281 032250  112722  000200                  MOVB    #200,(R2)+      ;TERMINATOR
2282 032254  004737  037360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032260  000404                          BR      220$            ;GO TO 220$ IF NO ERROR
     032262  000240                          NOP                     ;RETURN HERE IF ERROR
     032264  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032266  000137  032302                  JMP     230$            ;GO TO 230$ IF ERROR
2283 032272                                  220$:
2284 032272  000240                          NOP
2285 032274  004737  037722                  JSR     PC,TIMOUT       ;WAIT FOR TIME OUT
2286 032300  000240                          NOP
2287 032302                                  230$:
2288
```

```
                                        .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 'READY' ROUTINE.
                                        ;
032302  000004                  $EOSP:  SCOPE
032304  000240                          NOP
032306  013700  001464                  MOV     TSTQUE,R0       ;GET POINTER TO TSTQUE
032312  062700  000002                  ADD     #2,R0           ;ADJUST POINTER TO NEXT DEVICE
032316  010037  001464                  MOV     R0,TSTQUE       ;SAVE POINTER TO TSTQUE
032322  005710                          TST     (R0)            ;ANY MORE DEVICES FOR TEST ?
032324  001402                          BEQ     1$              ;BR IF NO
032326  000137  056756                  JMP     SHUT            ;JUMP TO SHUT AND CHECK FOR CONTROL C
032332  012737  001466  001464  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
032340                          $EOP:
032340  000240                          NOP
032342  005037  001116                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
032346  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
032352  005237  001230                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
032356  042737  100000  001230          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
032364  005327                          DEC     (PC)+           ;;LOOP?
032366  000001                  $EOPCT: .WORD   1
032370  003063                          BGT     $DOAGN          ;;YES
032372  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
032374  000001                  $ENDCT: .WORD   1
032376  032366                          $EOPCT
032400  104401  032406                  TYPE    ,65$            ;;TYPE ASCIZ STRING
032404  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
032424                          64$:
032424  013746  001230                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
032430  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
032432  104401  032440                  TYPE    ,67$            ;;TYPE ASCIZ STRING
032436  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
032502                          66$:
032502  013746  001126                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
032506  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
032510  104401  001217                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
032514  005037  001126                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
032520  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
```

```
032524  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
032526  000005                          RESET                   ;;CLEAR THE WORLD
032530  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
032532  000240                          NOP                     ;;SAVE ROOM
032534  000240                          NOP                     ;;FOR
032536  000240                          NOP                     ;;ACT11
032540                          $DOAGN:
032540  000137                          JMP     @(PC)+          ;;RETURN
032542  056756                  $RTNAD: .WORD   SHUT
032544     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        .EVEN
```

```
                                        .SBTTL  SUBROUTINES
                                        ;;******************************************************************
                                        .SBTTL  ERROR TYPEOUT ROUTINE

                                        ;*THE ERROR TYPEOUT ROUTINE ASSEMBLES AND PRINTS INFORMATION
                                        ;*REGARDING THE DETECTION OF AN ERROR AS FOLLOWS:
                                        ;*
                                        ;*      .UNIT NUMBER, TEST NUMBER, ERROR NUMBER AND PROGRAM COUNTER ARE
                                        ;*PRINTED ON THE FIRST LINE;
                                        ;*      .ERROR MESSAGE IS ASSEMELED, FORMATTED AND PRINTED ON
                                        ;*ONE OR MORE SUCCEEDING LINES;
                                        ;*      .PAIRED LINES OF ERROR HEADERS AND ERROR DATA
                                        ;*ARE PRINTED AFTER THE ERROR MESSAGE.

032550                                  ERRTYP:
032550  104414                                  SAVREG
032552  032777  020000  146374                  BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
032560  001402                                  BEQ     1$              ;NO!.
032562  000137  033300                          JMP     21$             ;YES!!
                                        ;TYPE UNIT NUMBER, TEST NUMBER, ERROR NUMBER, AND PROGRAM COUNTER
032566  104401  001217                1$:     TYPE    ,$CRLF
032572  104401  033314                        TYPE    ,ERTY00         ;TYPE "UNT#"
032576  013746  001234                        MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                        ;;TYPE UNIT NUMBER
032602  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
032604     003                                .BYTE   3               ;;TYPE 3 DIGIT(S)
032605     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
032606  005037  033304                        CLR     TSTNMB          ;LOAD TEST NUMBER FOR
032612  013737  001226  033304                MOV     $TESTN,TSTNMB
032620  104401  033322                        TYPE    ,ERTY01         ;TYPE "TST#"
032624  013746  033304                        MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                        ;;TYPE TEST NUMBER
032630  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
032632     003                                .BYTE   3               ;;TYPE 3 DIGIT(S)
032633     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
032634  005037  033306                        CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
032640  113737  001130  033306                MOVB    $ITEMB,ERRNMB   ;TYPEOUT
032646  001406                                BEQ     2$              ;SKIP IF NO ERROR CALLED
032650  104401  033332                        TYPE    ,ERTY02         ;TYPE "ERR#"
032654  013746  033306                        MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                        ;;TYPE ERROR NUMBER
032660  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
032662     003                                .BYTE   3               ;;TYPE 3 DIGIT(S)
032663     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
032664  104401  033341                2$:     TYPE    ,ERTY03         ;TYPE "PC="
032670  013746  001132                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;TYPE PROGRAM COUNTER
032674  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
032676     006                                .BYTE   6               ;;TYPE 6 DIGIT(S)
032677     001                                .BYTE   1               ;;TYPE LEADING ZEROS
                                        ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
032700  005737  033306                3$:     TST     ERRNMB          ;WAS AN ERROR CALLED?
032704  001575                                BEQ     21$             ;NO!!
032706  104401  001217                        TYPE    ,$CRLF          ;YES-TYPE CRLF
032712  105037  033312                        CLRB    BOTFLG          ;CLEAR BOT FLAG
032716  105037  033313                        CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
032722  013700  033306                        MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
```

```
                                        ASL     R0              ;FOUR ENTRIES IN ERROR
                                        ASL     R0              ;TABLE
                                        ASL     R0
                                        ADD     #$ERRTB-8.,R0
                                        MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
                                                                ;TABLE
                                        BEQ     13$             ;BRANCH IF NO ERROR MESSAGE
                                ;TYPE THE ERROR MESSAGE
                                4$:     MOV     (R1)+,R2        ;R2=ADDRESS OF MESSAGE STRING
                                        BEQ     12$             ;BRANCH IF END OF MESSAGE
                                        MOV     R2,11$          ;LOAD ADDRESS OF STRING
                                        CLR     BOTADR          ;CLEAR BOT ADDRESS
                                5$:     MOVB    (R2)+,R3        ;END OF STRING??
                                        BEQ     10$             ;YES!!
                                        CMPB    #CR,R3          ;CARRIAGE RETURN??
                                        BNE     6$              ;NO!!
                                        CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
                                        BR      5$              ;GET NEXT CHARACTER
                                6$:     CMPB    #LF,R3          ;LINE FEED??
                                        BEQ     5$              ;YES-GET NEXT CHARACTER
                                        CMPB    #HT,R3          ;HORIZONTAL TAB??
                                        BNE     8$              ;NO!!
                                7$:     INCB    CHRCNT          ;ADJUST CHARACTER COUNT
                                        BITB    #7,CHRCNT
                                        BNE     7$
                                        BR      9$
                                8$:     INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
                                        CMPB    #' ,R3          ;SPACE??
                                        BNE     9$              ;NO!!
                                        MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
                                9$:     CMPB    #64.,CHRCNT     ;END OF LINE??
                                        BHIS    5$              ;NO!!
                                        MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
                                        BNE     90$             ;BRANCH IF SPACE DETECTED
                                        TYPE    ,$CRLF          ;TYPE CRLF
                                        CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
                                        MOV     11$,R2          ;SET UP R2 FOR TESTING
                                        BR      5$
                                90$:    CLRB    -(R4)           ;REPLACE SPACE
                                        MOVB    #-1,BOTFLG      ;SET BOT FLAG
                                10$:    TYPE                    ;TYPE ERROR MESSAGE STRING
                                11$:    .WORD                   ;STRING ADDRESS GOES HERE
                                        TSTB    BOTFLG          ;WAS STRING TRUNCATED??
                                        BEQ     4$              ;NO!!
                                        TYPE    ,$CRLF          ;YES-TYPE CRLF
                                        CLRB    BOTFLG          ;CLEAR BOT FLAG
                                        CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
                                        MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
                                        MOV     R2,11$          ;SETUP 11$ FOR TYPING
                                        MOVB    #' ,-(R2)       ;RESTORE SPACE
                                        TSTB    (R2)+           ;RESTORE R2
                                        BR      5$              ;TYPE REST OF STRING
                                12$:
                                ;TYPE ERROR HEADER AND ERROR DATA
                                13$:
                                        MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
                                        BEQ     21$             ;BRANCH IF NO HEADER
```

```
 99 033170  104401  001217                  TYPE    ,$CRLF          ;(ASSUME NO DATA)
100 033174  016002  000004                  MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
101 033200  016003  000006                  MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
102 033204  012137  033214          14$:    MOV     (R1)+,15$       ;PUT HEADER ADDRESS FOR TYPE
103 033210  001433                          BEQ     21$             ;BRANCH IF END OF HEADERS
104                                                                 ;(ASSUME END OF DATA)
105 033212  104401                          TYPE
106 033214  000000          15$:    .WORD   0                       ;HEADER ADDRESS GOES HERE
107 033216  104401  001217                  TYPE    ,$CRLF
108 033222  005702                          TST     R2              ;DATA WITH HEADER??
109 033224  001767                          BEQ     14$             ;NO!!
110 033226  012204                          MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
111 033230  012305                          MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
112 033232  105725          16$:    TSTB    (R5)+                   ;WHAT KIND OF DATA??
113 033234  100407                          BMI     18$             ;BINARY
114 033236  001403                          BEQ     17$             ;OCTAL
115 033240  013446                          MOV     @(R4)+,-(SP)    ;DECIMAL
116 033242  104405                          TYPDS
117 033244  000405                          BR      19$
118 033246  013446          17$:    MOV     @(R4)+,-(SP)
119 033250  104402                          TYPOC
120 033252  000402                          BR      19$
121 033254  013446          18$:    MOV     @(R4)+,-(SP)
122 033256  104406                          TYPBN
123 033260  005714          19$:    TST     (R4)                    ;MORE DATA??
124 033262  001403                          BEQ     20$             ;NO!.
125 033264  104401  033347                  TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
126 033270  000760                          BR      16$             ;AND CONTINUE
127 033272  104401  001217          20$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
128 033276  000742                          BR      14$             ;BEFORE NEXT HEADER
129 033300  104415          21$:    RESREG
130 033302  000207                          RTS     PC
131
132 033304  000000          TSTNMB: .WORD   0                       ;TEST NUMBER
133 033306  000000          ERRNMB: .WORD   0                       ;ERROR NUMBER
134 033310  000000          BOTADR: .WORD   0                       ;BEGINNING OF TEXT ADDRESS
135 033312     000          BOTFLG: .BYTE   0                       ;BOT FLAG
136 033313     000          CHRCNT: .BYTE   0                       ;CHARACTER COUNT
137
138 033314     125     116     111  ERTY00: .ASCIZ  @UNIT#@
139 033322     054     040     124  ERTY01: .ASCIZ  @, TEST#@
140 033332     054     040     105  ERTY02: .ASCIZ  @, ERR#@
141 033341     054     040     120  ERTY03: .ASCIZ  @, PC=@
142 033347     040     040     000  ERTY04: .ASCIZ  @ @
143                                 .EVEN
144
```

```
                        .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   NNNNNN          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

033352                  TSTPRP:

                        ;STORE TASK DESCRIPTOR AND CLEAR USER'S ERROR CALL
033352  017637  000000  034272          MOV     @(SP),500$      ;STORE DESCRIPTOR
033360  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
033364  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR CALL
033370  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN

033374  004737  037024                  JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
033400  004737  037110                  JSR     PC,GET          ;GET RMER2
033404  000411                          BR      15$             ;BR IF NO ERROR DETECTED
033406  000401                          BR      10$             ;GET OVER ERROR NUMBER
033410  000000                          .WORD   0               ;ERROR DEFINED BY GET SUBROUTINE
033412  062716  000004          10$:    ADD     #4,(SP)         ;XFER ERROR TO USER AND
033416  113776  033410  000000        MOVB    10$-2,@(SP)     ;GET ERROR NUMBER.
033424  000137  034262                  JMP     400$

033430  013737  001376  034274  15$:    MOV     RMER2I,505$     ;GET RMER2 AND SAVE FOR LATER

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

```
                                ;SELECT DEVICE AND VERIFY DEVICE AVAILABLE IF BIT 15 SET IN TASK
033436  005737  034272                TST     500$            ;SELECT DEVICE??
033442  100014                        BPL     30$             ;NO!!

033444  004737  045012                JSR     PC,DEVSEL       ;GO SELECT DEVICE
033450  000411                        BR      30$             ;NO ERROR - CONTINUE
033452  000401                        BR      20$
033454  000000                        .WORD   0               ;ERROR NUMBER FROM DEVSEL
033456  062716  000004        20$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
033462  113776  033454  000000        MOVB    20$-2,@(SP)
033470  000137  034262                JMP     400$

                                ;;**************************************************************
                                ;CLEAR CONTROLLER IF BIT 14 IS SET IN TASK
033474                          30$:
033474  032737  040000  034272        BIT     #BIT14,500$     ;CLEAR CONTROLLER??
033502  001451                        BEQ     120$            ;NO!.

033504  004737  046464                JSR     PC,CNTCLR       ;GO CLEAR CONTROLLER
033510  000411                        BR      60$             ;CONTINUE - NO ERROR
033512  000401                        BR      50$
033514  000000                40$:    .WORD   0               ;ERROR NUMBER FROM CNTCLR
033516  062716  000004        50$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
033522  113776  033514  000000        MOVB    40$,@(SP)
033530  000137  034262                JMP     400$

                                ;;**************************************************************
                                ;VERIFY CONTROLLER CLEAR IF BIT6 SET IN TASK
033534                          60$:
033534  032737  000100  034272        BIT     #BIT6,500$      ;VERIFY??
033542  001431                        BEQ     120$            ;NO!!

033544  004737  037110                JSR     PC,GET          ;GO GET STATUS
033550  000411                        BR      90$             ;NO ERROR GETTING STATUS
033552  000401                        BR      80$
033554  000000                70$:    .WORD   0               ;ERROR FROM GETTING STATUS
033556  062716  000004        80$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
033562  113776  033554  000000        MOVB    70$,@(SP)
033570  000137  034262                JMP     400$

033574  004737  046602        90$:    JSR     PC,CLRSTS       ;GO VERIFY STATUS CLEAR
033600  000412                        BR      120$            ;NO ERROR IN CLEAR
033602  000401                        BR      110$
033604  000000                100$:   .WORD   0               ;ERROR IN STATUS CLEAR
033606  005726                110$:   TST     (SP)+           ;STRIP RETURN ADDRESS TO
033610  062716  000004                ADD     #4,(SP)         ;SUBROUTINE AND TRANSFER
033614  113776  033604  000000        MOVB    100$,@(SP)      ;ERROR TO USER
033622  000137  034262                JMP     400$

                                ;;**************************************************************
                                ;EXECUTE PACK ACKNOWLEDGE IF BIT12 SET IN TASK AND VOLUME IS
                                ;NOT VALID
033626                          120$:
033626  032737  010000  034272        BIT     #BIT12,500$     ;PACK ACKNOWLEDGE??
033634  001503                        BEQ     240$            ;NO..

033636  004737  037110                JSR     PC,GET
```

```
033642  000411                          BR      150$            ;NO ERROR GETTING RMDS
033644  000401                          BR      140$
033646  000000                  130$:   .WORD   0
033650  062716  000004         140$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
033654  113776  033646  000000         MOVB    130$,@(SP)
033662  000137  034262                 JMP     400$

033666  032737  000100  001346 150$:   BIT     #VV,RMDSI       ;IS VOLUME VALID??
033674  001063                         BNE     240$            ;YES!!

033676  005037  001510                 CLR     MEDENB          ;CLEAR MEDIA ENABLE
033702  012737  000023  001410         MOV     #PAKACK!GO,RMCS1O       ;LOAD PACK ACK COMMAND
033710  112737  000000  001551         MOVB    #RMCS1,PUTINX   ;SETUP REGISTER INDEX TABLE
033716  112737  000200  001552         MOVB    #200,PUTINX+1
033724  004737  037360                 JSR     PC,PUT          ;GO WRITE COMMAND
033730  000410                         BR      180$            ;NO ERROR LOADING REGISTER
033732  000401                         BR      170$
033734  000000                  160$:   .WORD   0               ;ERROR FROM PUT SUB
033736  062716  000004         170$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
033742  113776  033734  000000         MOVB    160$,@(SP)
033750  000544                         BR      400$

033752  004737  037722         180$:   JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

                                ;;*****************************************************************
                                ;VERIFY PACK ACKNOWLEDGE IF #BIT4 SET IN TASK
033756  032737  000020  034272         BIT     #BIT4,500$      ;VERIFY PACK ACKNOWLEDGE??
033764  001427                         BEQ     240$            ;NO!!

033766  004737  037110                 JSR     PC,GET          ;GO GET STATUS
033772  000410                         BR      210$            ;NO ERROR GETTING STATUS
033774  000401                         BR      200$
033776  000000                  190$:   .WORD   0               ;ERROR FROM GET SUB
034000  062716  000004         200$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
034004  113776  033776  000000         MOVB    190$,@(SP)
034012  000523                         BR      400$

034014  004737  047462         210$:   JSR     PC,ACKSTS       ;GO CHECK ACKNOWLEDGE
034020  000411                         BR      240$            ;NO ERROR
034022  000401                         BR      230$
034024  000000                  220$:   .WORD   0               ;PACK ACKNOWLEDGE ERROR
034026  005726                  230$:   TST     (SP)+           ;STRIP RETURN TO SUB AND
034030  062716  000004                 ADD     #4,(SP)         ;TRANSFER ERROR TO USER
034034  113776  034024  000000         MOVB    220$,@(SP)
034042  000507                         BR      400$

                                ;;*****************************************************************
                                ;RECALIBRATE DRIVE IF BIT 11 IS SET IN TASK AND "SKI" IS SET
                                ;OR "PIP" IS ACTIVE.
034044                          240$:
034044  032737  004000  034272         BIT     #BIT11,500$     ;RECALIBRATE??
034052  001505                         BEQ     410$            ;NO!!

034054  004737  037110                 JSR     PC,GET          ;GO GET RMDS
034060  000410                         BR      270$            ;NO ERROR GETTING RMDS
034062  000401                         BR      260$
034064  000000                  250$:   .WORD   0               ;ERROR FROM GET SUB
```

```
034066  062716  000004          260$:   ADD     #4,(SP)           ;TRANSFER ERROR TO USER
034072  113776  034064  000000          MOVB    250$,@(SP)
034100  000470                          BR      400$

034102  032737  040000  034274  270$:   BIT     #SKI,505$         ;WAS SKI SET ?
034110  001004                          BNE     280$              ;YES, GO RECALIBRATE
034112  032737  020000  001346          BIT     #PIP,RMDSI        ,IS PIP ACTIVE??
034120  001462                          BEQ     410$              ;NO!!

034122  012737  000007  001410  280$:   MOV     #RECAL!GO,RMCS10         ;LOAD RECALIBRATE COMMAND
034130  112737  000000  001551          MOVB    #RMCS1,PUTINX     ;AND REGISTER INDEX
034136  112737  000200  001552          MOVB    #200,PUTINX+1
034144  004737  037360                  JSR     PC,PUT            ;GO ISSUE RECALIBRATE
034150  000410                          BR      300$              ;NO ERROR
034152  000401                          BR      290$
034154  000000                          .WORD   0                 ;ERROR IN REGISTER TRANSFER
034156  062716  000004          290$:   ADD     #4,(SP)           ;TRANSFER ERROR TO USER
034162  113776  034154  000000          MOVB    290$-2,@(SP)
034170  000434                          BR      400$

034172  004737  037722          300$:   JSR     PC,TIMOUT         ;WAIT FOR COMPLETION

                                ;;*************************************************************
                                ;VERIFY RECALIBRATE IF BIT 3 SET IN TASK
034176  032737  000010  034272          BIT     #BIT3,500$        ;VERIFY RECALIBRATE??
034204  001430                          BEQ     410$              ;NO!!

034206  004737  037110                  JSR     PC,GET            ;GO GET STATUS
034212  000410                          BR      330$              ;NO ERROR GETTING STATUS
034214  000401                          BR      320$
034216  000000                  310$:   .WORD   0                 ;ERROR FROM GET
034220  062716  000004          320$:   ADD     #4,(SP)           ;TRANSFER ERROR TO USER
034224  113776  034216  000000          MOVB    310$,@(SP)
034232  000413                          BR      400$

034234  004737  050256          330$:   JSR     PC,RCLSTS         ;GO CHECK RECALIBRATE
034240  000412                          BR      410$              ;NO ERROR DURING RECALIBRATE
034242  000401                          BR      350$
034244  000000                  340$:   .WORD   0                 ;ERROR DURING RECALIBRATE
034246  005726                  350$:   TST     (SP)+             ;STRIP RETURN TO SUB AND
034250  062716  000004                  ADD     #4,(SP)           ;TRANSFER ERROR TO USER
034254  113776  034244  000000          MOVB    340$,@(SP)
034262  162716  000002          400$:   SUB     #2,(SP)           ;MOVE SP BACK BEFORE ERROR
034266  000240                  410$:   NOP
034270  000207                          RTS     PC                ;RETURN TO USER

034272  000000                  500$:   .WORD   0                 ;TASK/VERIFY DESCRIPTOR
034274  000000                  505$:   .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.
                        ;
034276                  BADSCT:
034276  062716  000006          ADD     #6,(SP)         ;CLEAR ERROR NUMBER IN USER'S
034302  105076  000000          CLRB    @(SP)           ;ERROR CALL.
034306  162716  000006          SUB     #6,(SP)
                        ;
                        ;TEST "MEDIA ENABLE" TO DETERMINE WHETHER OR NOT THE BAD SECTOR FILES
                        ;HAVE BEEN RECOVERED.
034312  005737  001510          TST     MEDENB          ;HAS BAD SECTOR FILES BEEN RECOVERED ?
034316  001402                  BEQ     5$              ;BR IF NO
034320  000137  035572          JMP     300$            ;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
034324                  5$:
034324  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
034326  005000                  CLR     R0              ;START WITH RMCS1
034330  016060  001410  101206  10$:    MOV     PUTBUF(R0),BUFFER(R0)
034336  062700  000002          ADD     #2,R0           ;ADVANCE TO NEXT BUFFER POSITION
034342  022700  000046          CMP     #46,R0          ;END OF BUFFER
034346  103370                  BHIS    10$             ;NO !!
                        ;
                        ;SET RETRY COUNT AND LOAD PUT BUFFER AND REGISTER INDEX TABLE
                        ;SETUP PARAMETERS TO READ SKIP SECTOR FILE FIRST (3 SECTORS)
034350  012737  000003  036210          MOV     #3,500$         ;RETRY COUNT
```

```
 034356  012737  001466  001444          MOV     #822.,RMDCO      ;DESIRED CYLINDER = 822.
 034364  013737  001332  001416          MOV     LSTRK,RMDAO      ;STARTING LAST TRACK, SECTOR = 0
 034372  012737  177376  001412          MOV     #-258.,RMWCO     ;2 + 256. WORDS (2'S COMP)
 034400  012737  010000  001442          MOV     #FMT16,RMOFO     ;16 BIT FORMAT
 034406  012737  103216  001414          MOV     #MFGFIL,RMBAO    ;POINT TO MANUFACTURES FILE BUFFER

 034414  012700  001551                  MOV     #PUTINX,RO       ;RO POINTS TO REGISTER INDEX TABLE
 034420  112720  000006                  MOVB    #RMDA,(RO)+
 034424  112720  000034                  MOVB    #RMDC,(RO)+
 034430  112720  000002                  MOVB    #RMWC,(RO)+
 034434  112720  000032                  MOVB    #RMOF,(RO)+
 034440  112720  000004                  MOVB    #RMBA,(RO)+
 034444  112720  000000                  MOVB    #RMCS1,(RO)+
 034450  112720  000200                  MOVB    #200,(RO)+
 034454  012600                          MOV     (SP)+,RO         ;.POP STACK INTO RO


                                 ;SET GET INDEX TABLE FOR READING STATUS
 034456                          20$:
 034456  004737  037024                  JSR     PC,GETSTS        ;SETUP GET INDEX REGISTER FOR STATUS
 034462  004737  037110                  JSR     PC,GET           ;GET REGISTERS
 034466  000411                          BR      30$              ;BR IF NO ERROR
 034470  000401                          BR      25$              ;JUMP OVER ERROR NUMBER
 034472  000000                          .WORD   0                ;ERROR DEFINED BY GET SUB
 034474  062716  000006          25$:    ADD     #6,(SP)          ;XFER ERROR TO USER AND
 034500  113776  034472  000000          MOVB    25$-2,@(SP)      ;GET ERROR NUMBER.
 034506  000137  035272                  JMP     215$

 034512  01373   001376  036222  30$:    MOV     RMER2I,550$      ;GET RMER2 AND SAVE FOR LATER


                                 ;CLEAR THE DEVICE USING DRIVE CLEAR COMMAND
 034520  012737  000011  001410          MOV     #DRVCLR!GO,RMCS10       ;LOAD COMMAND IN PUT BUFFER
 034526  004737  037360                  JSR     PC,PUT           ;OUTPUT COMMAND
 034532  000411                          BR      45$              ;RETURN HERE IF NO ERROR
 034534  000401                          BR      40$              ;GET AROUND ERROR #
 034536  000000                  35$:    .WORD   0                ;ERROR # GOES HERE
 034540  062716  000006          40$:    ADD     #6,(SP)          ;MOVE SP TO USERS ERROR CALL
 034544  113776  034536  000000          MOVB    35$,@(SP)        ;MOVE ERROR NUMBER TO USER
 034552  000137  035272                  JMP     215$

 034556  004737  037722          45$:    JSR     PC,TIMOUT        ;WAIT FOR COMPLETION
 034562  004737  037110                  JSR     PC,GET           ;GO GET STATUS
 034566  000411                          BR      60$              ;RETURN HERE IF NO ERROR
 034570  000401                          BR      55$              ;GET AROUND ERROR #
 034572  000000                  50$:    .WORD   0                ;ERROR # GOES HERE
 034574  062716  000006          55$:    ADD     #6,(SP)          ;MOVE SP TO USERS ERROR CALL
 034600  113776  034572  000000          MOVB    50$,@(SP)        ;MOVE ERROR # TO USERS ERROR CALL
 034606  000137  035272                  JMP     215$

 034612  004737  052020          60$:    JSR     PC,DRVSTS        ;GO VERIFY DRIVE CLEAR COMMAND
 034616  000412                          BR      75$              ;RETURN HERE IF NO ERROR
 034620  000401                          BR      70$              ;GET AROUND ERROR
 034622  000000                  65$:    .WORD   0                ;ERROR # GOES HERE
 034624  005726                  70$:    TST     (SP)+            ;STRIP RETURN TO SUBROUTINE
 034626  062716  000006                  ADD     #6,(SP)          ;MOVE SP TO USERS ERROR CALL
 034632  113776  034622  000000          MOVB    65$,@(SP)        ;MOVE ERROR # TO USER CALL
 034640  000137  035272                  JMP     215$

```

```
                                        ;ISSUE A PACK ACKNOWLEDGE IF VOLUME VALID IS RESET
                                        75$:
034644  032737  000100  001346          BIT     #VV,RMDSI       ;IS VV RESET ??
034652  001052                          BNE     120$            ;NO !!

034654  012737  000023  001410          MOV     #PACACK!GO,RMCS1O       ,LOAD COMMAND
034662  004737  037360                  JSR     PC,PUT          ;GO PUT COMMAND TO DRIVE
034666  000411                          BR      90$             ;RETURN HERE IF NO OUTPUT ERROR
034670  000401                          BR      85$             ;GET AROUND ERROR #
034672  000000                  80$:    .WORD   0               ;ERROR # GOES HERE
034674  062716  000006          85$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
034700  113776  034672  000000          MOVB    80$,@(SP)       ;MOVE ERROR # TO ERROR CALL
034706  000137  035272                  JMP     215$

034712  004737  037722          90$:    JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
034716  004737  037110                  JSR     PC,GET          ;GO GET STATUS FOR PACK ACK
034722  000411                          BR      105$            ;RETURN HERE IF NO ERROR
034724  000401                          BR      100$            ;GET AROUND ERROR #
034726  000000                  95$:    .WORD   0               ;ERROR # GOES HERE
034730  062716  000006          100$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
034734  113776  034726  000000          MOVB    95$,@(SP)       ;MOVE ERROR # TO CALL
034742  000137  035272                  JMP     215$

034746  004737  047462          105$:   JSR     PC,ACKSTS       ;GO VERIFY ACKNOWLEDGE STATUS
034752  000412                          BR      120$            ;RETURN HERE IF NO ERROR
034754  000401                          BR      115$            ;GET AROUND ERROR #
034756  000000                  110$:   .WORD   0               ;ERROR # GOES HERE
034760  005726                  115$:   TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
034762  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
034766  113776  034756  000000          MOVB    110$,@(SP)      ;MOVE ERROR # TO USERS ERROR CALL
034774  000137  035272                  JMP     215$

                                        ;RECALIBRATE THE DRIVE IF "SKI" OR "PIP IS SET
035000                                  120$:
035000  032737  040000  036222          BIT     #SKI,550$       ;WAS SKI SET ?
035006  001004                          BNE     125$            ;YES, GO RECALIBRATE
035010  032737  020000  001346          BIT     #PIP,RMDSI      ;IS PIP SET ??
035016  001452                          BEQ     165$            ;NO !!

035020  012737  000007  001410  125$:   MOV     #RECAL!GO,RMCS1O        ;LOAD RECALIBRATE COMMAND
035026  004737  037360                  JSR     PC,PUT          ;PUT THE RECAL COMMAND
035032  000411                          BR      135$            ;RETURN HERE IF NO ERROR
035034  000401                          BR      130$            ;GET AROUND ERROR #
035036  000000                          .WORD   0               ;ERROR # GOES HERE
035040  062716  000006          130$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035044  113776  035036  000000          MOVB    130$-2,@(SP)    ;MOVE ERROR # TO USERS CALL
035052  000137  035272                  JMP     215$

035056  004737  037722          135$:   JSR     PC,TIMOUT       ;WAIT FOR RECALIBRATE TO COMPLETE
035062  004737  037110                  JSR     PC,GET          ;GO GET RECAL STATUS
035066  000411                          BR      150$            ;RETURN HERE IF NO ERROR
035070  000401                          BR      145$            ;GET AROUND ERROR #
035072  000000                  140$:   .WORD   0               ;ERROR # GOES HERE
035074  062716  000006          145$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035100  113776  035072  000000          MOVB    140$,@(SP)      ;MOVE ERROR TO USERS CALL
035106  000137  035272                  JMP     215$
```

```
                                150$:   JSR     PC,RCLSTS       ;GO VERIFY RECALIBRATE STATUS
                                        BR      165$            ;RETURN HERE IF NO ERROR
                                        BR      160$            ;GET AROUND ERROR #
                                155$:   .WORD   0               ;ERROR # GOES HERE
                                160$:   TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
                                        ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    155$,@(SP)      ;MOVE ERROR # TO USERS CALL
                                        JMP     215$

                                ;READ THE SECTOR IDENTIFIED BY RMDAO, INCLUDING HEADER AND DATA
                                165$:
                                        MOV     #RH!GO,RMCS1O   ;LOAD READ HEADER AND DATA COMMAND
                                        JSR     PC,PUT          ;PUT COMMAND
                                        BR      180$            ;RETURN HERE IF NO ERROR
                                        BR      175$            ;GET AROUND ERROR #
                                170$:   .WORD   0               ;ERROR # GOES HERE
                                175$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    170$,@(SP)      ;MOVE ERROR # TO USERS ERROR CALL
                                        JMP     215$

                                180$:   JSR     PC,TIMOUT       ;WAIT FOR READ OPERATION TO COMPLETE
                                        JSR     PC,GET          ;GO GET STATUS FOR READ OPERATION
                                        BR      195$            ;RETURN HERE IF NO ERROR
                                        BR      190$            ;GET AROUND ERROR #
                                185$:   .WORD   0               ;ERROR # GOES HERE
                                190$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    185$,@(SP)      ;MOVE ERROR # TO CALL
                                        JMP     215$

                                195$:   JSR     PC,DTASTS       ;GO VERIFY RESULTS OF READ OPERATION
                                        BR      210$            ;RETURN HERE IF NO ERROR
                                        BR      205$            ;GET AROUND ERROR #
                                200$:   .WORD   0               ;ERROR # GOES HERE
                                205$:   TST     (SP)+           ;STRIP RETURN ADDRESS TO SUBROUTINE
                                        ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    200$,@(SP)      ;MOVE ERROR # TO USERS CALL
                                        JMP     215$

                                210$:   BR      240$            ;NO ERRORS DETECTED

                                ;;*****************************************************************
                                ;AN ERROR HAS BEEN DETECTED IN TRYING TO READ THE BAD SECTOR FILE.
                                ;THE SECTOR WILL BE RETRIED IF POSSIBLE.
                                215$:
                                        DEC     500$            ;YES, DECREMENT RETRY COUNT AND
                                        BPL     225$            ;RETRY IF COUNT NOT NEGATIVE.

                                ;THE RETRY COUNT HAS EXPIRED - SEE IF THE ERROR IS MEDIA RELATED
                                220$:
                                        MOV     RMER2I,-(SP)    ;GET ER2
                                        BIC     #BSE,(SP)+      ;ANY NON-MEDIA ERRORS ?
                                        BNE     230$            ;YES, EXIT AND REPORT ERROR ON RETURN

                                        MOV     RMER1I,-(SP)    ;GET ER1
                                        BIC     #DCK!HCRC!HCE!FER!ECH,(SP)+
                                                                ;ARE THERE ANY NON-MEDIA ERRORS ?
                                        BNE     230$            ;YES, 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
035324  062737  000002  001416          ADD     #2,RMDAO        ;ADVANCE SECTOR ADDRESS BY 2
035332  122737  000012  001416          CMPB    #10.,RMDAO      ;QUIT IF ALL MFG SECTORS HAVE BEEN
035340  001413                          BEQ     230$            ;TRIED.
035342  122737  000040  001416          CMPB    #32.,RMDAO      ;QUIT IF ALL USER SECTORS HAVE BEEN
035350  001407                          BEQ     230$            ;TRIED.

035352  012737  000003  036210          MOV     #3,500$         ;REINSTATE RETRY COUNT FOR THIS SECTOR
035360  162716  000006          225$:   SUB     #6,(SP)         ;MOVE SP BACK TO NO ERROR
035364  000137  034456                  JMP     20$             ;RETRY THE READ OPERATION

                                        ;THE BAD SECTOR FILE CANNOT BE READ
035370                          230$:
035370  000240                          NOP
035372  032777  020000  143554          BIT     #SW13,@SWR      ;INHIBIT MESSAGE ?
035400  001002                          BNE     235$            ;YES
035402  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO ERROR RETURN
035406  000137  036204          235$:   JMP     410$            ;GO TO MODULE EXIT

                                        ;THE SECTOR WAS RECOVERED WITHOUT ERROR - READ THE USER FILE IF
                                        ;THIS IS THE MGF FILE OR ELSE DONE.
035412                          240$:
035412  022737  104224  001414          CMP     #USRFIL,RMBAO   ;WAS THE USER FILE READ ??
035420  001446                          BEQ     260$            ;YES - READ IS COMPLETE
035422  112737  000012  001416          MOVB    #10.,RMDAO      ;READ THE USER FILE LAST TRACK, SECTOR = 10.
035430  012737  104224  001414          MOV     #USRFIL,RMBAO   ;POINT TO USERS FILE BUFFER

035436  012737  000003  036210          MOV     #3,500$         ;RELOAD THE RETRY COUNT FOR THIS SECTOR
035444  000137  034456                  JMP     20$             ;GO READ THE USER FILE

                                        ;DUMMY THE BAD SECTOR FILES
035450                          250$:
035450  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
035452  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
035454  012701  000374                  MOV     #252.,R1        ;R1 = NUMBER OF ENTRIES IN FILES
035460  012700  000014                  MOV     #14,R0          ;R0 = ADDRESS INDEX TO FILE  STORAGE
035464  012760  177777  103216  255$:   MOV     #-1,MFGFIL(R0)  ;ENTER ALL ONES IN MFG FILE
035472  012760  177777  104224          MOV     #-1,USRFIL(R0)  ;ENTER ALL ONES IN USER FILE
035500  005720                          TST     (R0)+           ;ADVANCE ADDRESS
035502  005301                          DEC     R1              ;DECREMENT COUNT
035504  001367                          BNE     255$            ;CONTINUE IF NOT DONE

035506  012701  000006                  MOV     #6.,R1          ;CLEAR HEADER, CLEAR ID & SERIAL NUMBERS
035512  005000                          CLR     R0
035514  005060  103216          257$:   CLR     MFGFIL(R0)
035520  005060  104224                  CLR     USRFIL(R0)
035524  005720                          TST     (R0)+           ;ADVANCE ADDRESS
035526  005301                          DEC     R1
035530  001371                          BNE     ^  $
035532  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
035534  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0

                                        ;SET MEDIA ENABLE AND RESTORE THE USERS PUT BUFFER
```

```
                                260$:
035536  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
035540  005000                          CLR     R0              ;R0 IS REGISTER INDEX
035542  012737  177777  001510          MOV     #-1,MEDENB
035550  016060  101206  001410  265$:   MOV     BUFFER(R0),PUTBUF(R0)
035556  062700  000002                  ADD     #2,R0           ;ADVANCE R0
035562  022700  000046                  CMP     #46,R0          ;DONE ??
035566  103370                          BHIS    265$
035570  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0

                                ;;*****************************************************************
                                ;VERIFY THAT THE DESIRED SECTOR IS NOT IN THE MFG BAD SECTOR FILE
                                ;AND NOT IN THE USERS BAD SECTOR FILE. ASSIGN A NEW SECTOR IF THE
                                ;SECTOR IS IN ANY OF THE FILES.
                                ;;*****************************************************************

                                ;LOAD INITIAL VARIABLES AND COMPUTE THE NUMBER OF SECTORS
035572                          300$:
035572  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
035574  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
035576  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
035600  013737  001444  001512          MOV     RMDCO,ASNDC     ;LOAD REQUESTED CYLINDER, TRACK,
035606  013737  001416  001514          MOV     RMDAO,ASNDA     ;AND SECTOR ADDRESS IN ASSIGNED STORAGE
035614  005002                          CLR     R2              ;R2 = NUMBER OF SECTORS
035616  013700  001412                  MOV     RMWCO,R0        ;R0 = WORD COUNT
035622  005400                          NEG     R0              ;MAKE NUMBER POSITIVE
035624  012701  000400                  MOV     #256.,R1        ;R1 = NUMBER OF WORDS PER SECTOR
035630  032737  000002  001410          BIT     #BIT1,RMCS10    ;IS THIS A HEADER AND DATA COMMAND ??
035636  001402                          BEQ     305$            ;NO !!
035640  012701  000402                  MOV     #258.,R1        ;CHANGE WORDS PER SECTOR
035644  020100                  305$:   CMP     R1,R0           ;IS THERE A FULL SECTOR ??
035646  101404                          BLOS    310$            ;YES !!
035650  005700                          TST     R0              ;IS R0 ZERO ??
035652  001405                          BEQ     315$            ;YES !!
035654  005202                          INC     R2              ;INCREMENT FOR PARTIAL SECTOR
035656  000403                          BR      315$
035660  160100                  310$:   SUB     R1,R0           ;SUBTRACT ONE SECTOR FROM WORD COUNT
035662  005202                          INC     R2              ;INCREMENT SECTOR COUNT
035664  000767                          BR      305$
035666  010237  036210          315$:   MOV     R2,500$         ;SAVE SECTOR COUNT

                                ;LOAD PARAMETERS AND SEARCH THE MFG/USER SECTOR FILE FOR THE
                                ;ASSIGNED SECTOR. ALSO, SEARCH THE ADJACENT SECTORS IF THE
                                ;SECTOR COUNT IS MORE THAN ONE.

035672  012737  103232  036220          MOV     #MFGFIL+14,540$ ;THE STARTING ADDRESS OF MFG FILE
                                                                ;TO BASE ADDRESS STORAGE.
035700  004737  035722                  JSR     PC,320$         ;GO SEARCH FILE
035704  012737  104240  036220          MOV     #USRFIL+14,540$ ;LOAD STARTING ADDRESS OF USR FILE
                                                                ;TO BASE ADDRESS STORAGE.
035712  004737  035722                  JSR     PC,320$         ;GO SEARCH FILE
035716  000137  036162                  JMP     400$            ;DONE WITH ALL FILE SEARCHES !.

035722  013737  001512  036214  320$:   MOV     ASNDC,520$      ;LOAD COMPARING CYLINDER ADDRESS
035730  013737  001514  036216          MOV     ASNDA,530$      ;LOAD COMPARING TRACK, SECTOR ADDRESS
035736  013737  036210  036212          MOV     500$,510$       ;LOAD NUMBER OF SECTORS TO CONFIRM
```

```
                                        ;SETUP FOR A BINARY SEARCH OF THE CURRENT FILE FOR THE COMPARING
                                        ;CYLINDER, TRACK AND SECTOR ADDRESS
036744                                  325$:
036744  013700  036220                          MOV     540$,R0         ;LOAD THE BASE ADDRESS IN R0
036750  022710  177777                  330$:   CMP     #-1,(R0)        ;IS THIS FILE TERMINATOR ?
036754  001446                                  BEQ     370$            ;BR IF YES
036756  021037  036214                          CMP     (R0),520$       ;DOES TABLE ENTRY = COMPARING CYLINDER ?
036762  001010                                  BNE     350$            ;BR IF NO

                                        ;FILE ENTRY EQUALS COMPARING CYLINDER.  SEE IF THE NEXT ENTRY EQUALS
                                        ;THE COMPARING TRACK, AND SECTOR.
036764                                  345$:
036764  126037  000003  036217                  CMPB    3(R0),530$+1    ;DOES TABLE ENTRY = COMPARING TRACK ?
036772  001004                                  BNE     350$            ;BR IF NO

036774  126037  000002  036216                  CMPB    2(R0),530$      ;DOES TABLE ENTRY = COMPARING SECTOR ?
036002  001402                                  BEQ     360$            ;BR IF YES

036004  022020                          350$:   CMP     (R0)+,(R0)+     ;NO, ADJUST CYLINDER POINTER IN BAD FILE
036006  000760                                  BR      330$            ;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.
036010                                  360$:
036010  105237  001514                          INCB    ASNDA           ;INCREMENT SECTOR
036014  122737  000037  001514                  CMPB    #31.,ASNDA      ;SECTOR OK ??
036022  103337                                  BHIS    320$            ;YES !!
036024  105037  001514                          CLRB    ASNDA           ;CLEAR SECTOR AND ADVANCE TRACK
036030  105237  001515                          INCB    ASNDA+1
036034  123737  001333  001515                  CMPB    LSTRK+1,ASNDA+1 ;TRACK OK ?
036042  103327                                  BHIS    320$            ;YES !!
036044  005037  001514                          CLR     ASNDA           ;CLEAR TRACK AND SECTOR
036050  005237  001512                          INC     ASNDC           ;INCREMENT CYLINDER
036054  022737  001466  001512                  CMP     #822.,ASNDC     ;CYLINDER OK ??
036062  103317                                  BHIS    320$            ;YES !!
036064  005037  001512                          CLR     ASNDC           ;START AT CYLINDER 0
036070  000714                                  BR      320$            ;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.
036072                                  370$:
036072  005337  036212                          DEC     510$            ;DECREMENT NUMBER OF SECTORS TO COMPARE
036076  001442                                  BEQ     410$            ;DONE IF ZERO

036100  105237  036216                          INCB    530$            ;INCREMENT THE COMPARING SECTOR
036104  122737  000037  036216                  CMPB    #31.,530$       ;SECTOR OK ??
036112  103022                                  BHIS    375$            ;YES !!
036114  105037  036216                          CLRB    530$            ;CLEAR SECTOR
036120  105237  036217                          INCB    530$+1          ;INCREMENT TRACK
036124  123737  001333  036217                  CMPB    LSTRK+1,530$+1  ;TRACK OK ??
036132  103012                                  BHIS    375$            ;YES !!
036134  005037  036216                          CLR     530$            ;CLEAR SECTOR TRACK
036140  005237  036214                          INC     520$            ;INCREMENT CYLINDER
036144  022737  001466  036214                  CMP     #822.,520$      ;CYLINDER OK ??
036152  103002                                  BHIS    375$            ;YES !!
036154  005037  036214                          CLR     520$            ;START AT CYLINDER 0
```

```
                                375$:   BR      325$            ;SEARCH NEXT SECTOR

                                ;ASSIGN THE SECTOR AND RETURN TO USER
036162                          400$:
036162  013737  001512  001444        MOV     ASNDC,RMDCO     ;LOAD CYLINDER
036170  013737  001514  001416        MOV     ASNDA,RMDAO     ;LOAD TRACK AND SECTOR
036176  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
036200  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
036202  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0

036204  000240                410$:   NOP
036206  000207                        RTS     PC

                                ;THE FOLLOWING ARE STORAGE LOCATIONS FOR THE MODULE

036210  000000                500$:   .WORD   0       ;RETRY COUNT/ NUMBER OF SECTORS REQUIRED
036212  000000                510$:   .WORD   0       ;NUMBER OF SECTORS TO COMPARE
036214  000000                520$:   .WORD   0       ;COMPARING CYLINDER
036216  000000                530$:   .WORD   0       ;COMPARING TRACK AND SECTOR
036220  000000                540$:   .WORD   0       ;BASE ADDRESS OF BAD SECTOR FILE BEING SEARCHED
036222  000000                550$:   .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

 036224                         GENBUF:
 036224  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 036226  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 036230  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 036232  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 036234  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 036236  013700  001414                 MOV     RMBA0,R0        ;LOAD DATA BUFFER ADDRESS
 036242  013701  001412                 MOV     RMWC0,R1        ;LOAD WORD COUNT
 036246  013737  001444  036456         MOV     RMDC0,60$       ;LOAD STARTING CYLINDER ADDRESS
 036254  013737  001416  036460         MOV     RMDA0,65$       ;LOAD STARTING TRACK,SECTOR ADDRESS

 036262  032737  000002  001410 10$:    BIT     #BIT1,RMCS10    ;WRITE HEADER & DATA??
 036270  001445                         BEQ     25$             ;NO!!
 036272  013710  036456                 MOV     60$,(R0)        ;WRITE HEADER WORD #1
 036276  052710  140000                 BIS     #MSE!USE,(R0)   ;SET BAD SECTOR FLAGS FOR GOOD SECTOR
 036302  012702  000035                 MOV     #29.,R2         ;R2 = MAXIMUM SECTOR ADDRESS (29.)

 036306  032737  010000  001442         BIT     #FMT16,RMOF0    ;18 BIT FORMAT??
 036314  001404                         BEQ     15$             ;YES !!
 036316  052710  010000                 BIS     #FMT16,(R0)     ;SET 16 FORMAT BIT IN HEADER
 036322  012702  000037                 MOV     #31.,R2         ;CHANGE MAXIMUM SECTOR ADDRESS (31.)

 036326  005201                 15$:    INC     R1              ;INCREMENT WORD COUNT
 036330  001443                         BEQ     50$             ;EXIT IF DONE

 036332  005720                         TST     (R0)+           ;MOVE R0 TO HEADER WORD #2
 036334  013720  036460                 MOV     65$,(R0)+       ;WRITE HEADER WORD #2
 036340  005201                         INC     R1              ;INCREMENT WORD COUNT AND
 036342  001436                         BEQ     50$             ;EXIT IF DONE
 036344  012703  036460                 MOV     #65$,R3         ;ADVANCE SECTOR ADDRESS
 036350  105213                         INCB    (R3)
 036352  120213                         CMPB    R2,(R3)         ;SECTOR OVERFLOW ??
 036354  103013                         BHIS    25$             ;NO !!
 036356  105013                         CLRB    (R3)            ;YES - CLEAR SECTOR ADDRESS
 036360  105263  000001                 INCB    1(R3)           ;ADVANCE TRACK ADDRESS
 036364  123763  001333  000001         CMPB    LSTRK+1,1(R3)   ;TRACK OVERFLOW ??
 036372  103004                         BHIS    25$             ;NO !!
 036374  105063  000001                 CLRB    1(R3)           ;YES - CLEAR TRACK ADDRESS
```

```
 53 036400  105237  036456                  INCB    60$             ;ADVANCE CYLINDER ADDRESS
 54 036404  012704  000400          25$.    MOV     #256.,R4        ;LOAD SECTOR DATA COUNT
 55 036410  013702  001174          30$:    MOV     $TMP0,R2        ;LOAD PATTERN ADDRESS
 56 036414  013703  001176                  MOV     $TMP1,R3        ;LOAD PATTERN COUNT
 57 036420  012220                  40$:    MOV     (R2)+,(R0)+     ;WRITE DATA PATTERN
 58 036422  005201                          INC     R1              ;INCREMENT WORD COUNT AND
 59 036424  001405                          BEQ     50$             ;EXIT IF DONE
 60 036426  005304                          DEC     R4              ;DECREMENT SECTOR COUNT
 61 036430  001714                          BEQ     10$             ;START NEXT SECTOR IF 0
 62 036432  005303                          DEC     R3              ;DECREMENT PATTERN COUNT
 63 036434  001765                          BEQ     30$             ;RESTART PATTERN IF 0
 64 036436  000770                          BR      40$             ;CONTINUE DATA PATTERN
 65 036440                          50$:
    036440  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
    036442  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
    036444  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
    036446  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
    036450  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 66 036452  000240                          NOP
 67 036454  000207                          RTS     PC
 68
 69 036456  000000                  60$:    .WORD                   ;CYLINDER ADDRESS STORAGE
 70 036460  000000                  65$:    .WORD                   ;TRACK, SECTOR ADDRESS STORAGE
 71
```

```
                                .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)    ???
                                ;
036462                          CMPBUF:
036462  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
036464  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
036466  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
036470  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
036472  005037  037022                  CLR     150$            ;CLEAR CORRECTION FLAG
                                ;
                                ;DETERMINE IF DATA SHOULD BE CORRECTED
036476  033737  004000  001366          BIT     ECI,RMOFI       ;WAS ECC CORRECTION ALLOWED ??
036504  001063                        BNE     80$             ;NO !!
036506  032737  100000  001350        BIT     #DCK,RMER1I     ;WAS THERE A DATA CHECK ??
036514  001457                        BEQ     80$             ;NO !!
036516  032737  000100  001350        BIT     #ECH,RMER1I     ;IS ERROR CORRECTION HARD SET ?
036524  001053                        BNE     80$             ;YES !!
036526  032737  010000  001366        BIT     #FMT16,RMOFI    ;IS THIS 16 BIT FORMAT ??
036534  001447                        BEQ     80$             ;NO !!
                                ;
                                ;CORRECT DATA USING ECC INFORMATION
036536  013700  001414                  MOV     RMBAO,R0        ;R0 = STARTING BUFFER ADDRESS
036542  013701  001400                  MOV     RMEC1I,R1       ;R1 = ECC POSITION
036546  052737  100000  037022          BIS     #BIT15,150$     ;SET CORRECTION FLAG
                                ;
                                ;MOVE R0 TO WORD BOUNDARY OF ERROR BURST
036554  022701  000020          10$:    CMP     #16.,R1         ;IS BIT POSITION > 1 WORD
036560  103004                        BHIS    20$             ;NO !!
036562  162701  000020                  SUB     #16.,R1         ;SUBTRACT 1 WORDS WORTH
036566  005720                        TST     (R0)+           ;ADVANCE BUFFER ADDRESS 1 WORD
036570  000771                        BR      10$
036572  012702  000001          20$:    MOV     #1,R2           ;R2 = BIT POINTER
036576  010203                        MOV     R2,R3           ;R3 = BIT NUMBER
                                ;
                                ;MOVE R2 TO STARTING BIT OF ERROR BURST
036600  020301                  30$:    CMP     R3,R1           ;IS R3 SAME AS R1 ??
036602  001403                        BEQ     35$             ;YES !!
036604  006302                        ASL     R2              ;SHIFT BIT POINTER
036606  005203                        INC     R3              ;INCREMENT BIT NUMBER
036610  000773                        BR      30$
036612  012703  000013          35$:    MOV     #11.,R3         ;R3 = LENGTH OF ERROR BURST
                                ;
                                ;CORRECT THE ERROR BURST
036616  030237  001402          40$:    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$:    MOV     @10(SP),R0      ;R0 = WRITE BUFFER
                                        ADD     #2,10(SP)       ;MOVE SP TO READ ADDRESS
                                        MOV     @10(SP),R1      ;R1 = READ BUFFER
                                        ADD     #2,10(SP)       ;MOVE SP TO RETURN ADDRESS
                                        MOV     RMWC1,R2        ;R2 = NUMBER OF WORDS TRANSFER
                                        SUB     RMWC0,R2
                                90$:    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 COMPARE 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

037024                  GETSTS:
037024  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037026  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037030  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037032  012700  001522         MOV     #GETINX,R0      ;R0 = ADDRESS OF INDEX TABLE
037036  012701  001404         MOV     #RMEC21+2,R1    ;R1 = ADDRESS OF GET BUFFER
037042  012702  000046         MOV     #RMEC2,R2       ;R2 = REGISTER INDE
037046  110220         2$:     MOVB    R2,(R0)+        ;WRITE REGISTER INDEX IN TABLE
037050  005041                 CLR     -(R1)           ;CLEAR CORRESPONDING LOCATION
037052  162702  000002 3$:     SUB     #2,R2           ;DECREMENT TO NEXT INDEX
037056  100405                 BMI     4$              ;BRANCH OUT IF DONE
037060  022702  000022         CMP     #RMDB,R2        ;DONT WRITE RMDB INDEX
037064  001370                 BNE     2$
037066  005041                 CLR     -(R1)
037070  000770                 BR      3$
037072  112720  000200 4$.     MOVB    #200,(R0)+      ;WRITE TERMINATOR
037076  012602                 MOV     (SP)+,R2        ;;POP STACK INTO R2
037100  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
037102  012600                 MOV     (SP)+,R0        ;;POP STACK INTO R0
037104  000240                 NOP
037106  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

037110  000240          GET:    NOP
037112  062716  000004          ADD     #4,(SP)         ;CLEAR ERROR NUMBER IN USER'S
037116  105076  000000          CLRB    @(SP)           ;ERROR CALL
037122  162716  000004          SUB     #4,(SP)
037126  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037130  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037132  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037134  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
037136  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
037140  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
037144  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
037150  013700  001276          MOV     $BASE,R0
037154  012702  001334          MOV     #GETBUF,R2
037160  012704  001522          MOV     #GETINX,R4
037164  012737  037272  000004  MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
037172  012737  000300  000006  MOV     #PR6,ERRVEC+2
037200  016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
037206  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
037214  032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
037222  001007                  BNE     3$              ;YES!!
037224  062766  000004  000016  ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN USER'S
037232  112776  000112  000016  MOVB    #112,@16(SP)    ;ERROR CALL
037240  000423                  BR      7$
037242  105714          3$:     TSTB    (R4)            ;DONE??
037244  100433                  BMI     9$              ;YES!!
037246  111401                  MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
037250  042701  177700          BIC     #^C<IDXMSK>,R1  ;CLEAR ANY SIGN EXTENSION
```

```
     52 037254  060001                          ADD     R0,R1
     53 037256  112403                          MOVB    (P4)+,R3        ;R3 = STORAGE ADDRESS FOR REGISTER
     54 037260  042703  177700                  BIC     #^CIDXMSK,R3    ;CLEAR ANY SIGN EXTENSION
     55 037264  060203                          ADD     R2,R3
     56 037266  011113                          MOV     (R1),(R3)       ;READ REGISTER
     57 037270  000764                          BR      3$
     58
     59 037272  022626                  5$:     CMP     (SP)+,(SP)+     ;RESTORE STACK
     60 037274  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     61 037302  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     62 037310  162766  000002  000016  7$:     SUB     #2,16(SP)
     63 037316  105714                  8$:     TSTB    (R4)            ;DONE CLEARING??
     64 037320  100405                          BMI     9$              ;YES!!
     65 037322  005003                          CLR     R3              ;CLEAR REMAINING STORAGE
     66 037324  112403                          MOVB    (R4)+,R3        ;LOCATIONS
     67 037326  060203                          ADD     R2,R3
     68 037330  005013                          CLR     (R3)
     69 037332  000771                          BR      8$
     70 037334                          9$:
        037334  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        037340  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        037344  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        037346  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        037350  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        037352  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        037354  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     71 037356  000207                          RTS     PC              ;RETURN
     72
```

```
                                .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
                                ;
037360  000240          PUT:   NOP
037362  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037364  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037366  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037370  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
037372  010446                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
037374  013746  000004         MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
037400  013746  000006         MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
037404  013700  001276         MOV     $BASE,R0
037410  012702  001410         MOV     #PUTBUF,R2
037414  012704  001551         MOV     #PUTINX,R4
037420  012737  037530  000004 MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
037426  012737  000300  000006 MOV     #PR6,ERRVEC+2
037434  016037  000010  001174 1$:    MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
037442  016037  000000  001176 MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
037450  032737  004000  001176 BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
037456  001007                 BNE     3$              ;YES!!
037460  062766  000004  000016 ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
037466  112776  000112  000016 MOVB    #112,@16(SP)    ;USER'S ERROR CALL
037474  000424                 BR      7$
037476  105714          3$:    TSTB    (R4)            ;DONE??
037500  100425                 BMI     9$              ;YES!!
037502  111401                 MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
037504  042701  177700         BIC     #^C<IDXMSK>,R1  ;CLEAR ANY SIGN EXTENSION
037510  060001                 ADD     R0,R1
037512  111403                 MOVB    (R4),R3         ;R3 = STORAGE ADDRESS
037514  042703  177700         BIC     #^C<IDXMSK>,R3  ;CLEAR ANY SIGN EXTENSION
037520  060203                 ADD     R2,R3
037522  011311                 MOV     (R3),(R1)       ;WRITE REGISTER
037524  105724          4$:    TSTB    (R4)+           ;ADJUST REGISTER POINTER
```

```
     52 037526  000763                          BR      3$
     53
     54 037530  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
     55 037532  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     56 037540  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     57 037546  162766  000002  000016  7$:     SUB     #2,16(SP)
     58
     59 037554                          9$:
        037554  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        037560  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        037564  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        037566  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        037570  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        037572  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        037574  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     60 037576  000207                          RTS     PC              ;RETURN
     61
```

```
                                        .SBTTL  SIZE CLOCK SUBROUTINE

037600                                  SIZCLK:
037600  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
037604  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
037610  012737  037646  000004                  MOV     #1$,ERRVEC      ;SET UP FOR BUS TIMEOUT
037616  012737  000300  000006                  MOV     #PR6,ERRVEC+2
037624  012737  177546  001516                  MOV     #177546,CLKADR  ;LOAD ADDRESSES FOR KW11-L
037632  012737  000100  001520                  MOV     #100,CLKVCT
037640  005777  141652                          TST     @CLKADR         ;TEST FOR KW11-L PRESENT
037644  000421                                  BR      3$              ;YES - KW11-L IS PRESENT
037646  022626                          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP
037650  012737  037700  000004                  MOV     #2$,ERRVEC      ;SET UP FOR BUS TIMEOUT
037656  012737  172540  001516                  MOV     #172540,CLKADR  ;LOAD ADDRESSES FOR KW11-P CLOCK
037664  012737  000104  001520                  MOV     #104,CLKVCT
037672  005777  141620                          TST     @CLKADR         ;TEST FOR KW11-P PRESENT
037676  000404                                  BR      3$              ;YES - KW11-P IS PRESENT
037700  022626                          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP
037702  062766  000002  000004                  ADD     #2,4(SP)        ;MOVE RETURN TO ERROR
037710                                  3$:
037710  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
037714  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
037720  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
                                        ;
 037722                                 TIMOUT:
 037722  010046                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 037724  010146                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 037726  010246                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 037730  013746  000004                         MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
 037734  013746  000006                         MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
 037740  012737  040042  000004                 MOV     #4$,ERRVEC      ;SETUP FOR BUS TIMEOUT - 04 TRAP
 037746  012737  000300  000006                 MOV     #PR6,ERRVEC+2
 037754  013700  001276                         MOV     $BASE,R0        ;R0=BASE ADDRESS
 037760  013701  001516                         MOV     CLKADR,R1       ;R1=CLOCK ADDRESS
 037764  012702  000036                         MOV     #30.,R2         ;R2=NUMBER OF CLOCK CYCLES
 037770  020127  172540          1$:    CMP     R1,#172540      ;KW11-P CLOCK??
 037774  001003                         BNE     2$              ;NO!!
 037776  012761  000001  000002                 MOV     #1,2(R1)        ;SET COUNTER
 040004  012711  000005          2$:    MOV     #BIT2!BIT0,(R1) ;START COUNTER
                                        ;
 040010  016046  000000          3$:    MOV     RMCS1(R0),-(SP) ;GET STATUS
 040014  042716  177576                 BIC     #^C<RDY!GO>,(SP)
 040020  022726  000200                 CMP     #RDY,(SP)+      ;RDY=1,GO=0??
 040024  001420                         BEQ     5$              ;YES!!
 040026  032711  000200                 BIT     #BIT7,(R1)      ;TIMER DONE??
 040032  001766                         BEQ     3$              ;NO!!
 040034  005302                         DEC     R2              ;DEC NUMBER OF CYCLES
 040036  001354                         BNE     1$              ;CONTINUE IF NOT DONE
 040040  000412                         BR      5$              ;"RDY" DID NOT SET OR "GO" DID NOT RESET
                                                                ;WITHIN 500 MSEC AFTER THE COMMAND WAS ISSUED.
 040042  022626                 4$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
 040044  062766  000004  000012         ADD     #4,12(SP)       ;MOVE SP TO USER'S CALL
 040052  112776  000007  000012         MOVB    #7,@12(SP)      ;WRITE ERROR NUMBER
 040060  162766  000002  000012         SUB     #2,12(SP)
                                        ;
 040066                         5$:
 040066  012637  000006                 MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
 040072  012637  000004                 MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
 040076  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
 040100  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 040102  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 040104  000207                         RTS     PC              ;RETURN TO USER
                                        ;
```

```
                        .SBTTL  ERROR CHECK SUBROUTINES
                        ;;*****************************************************************
                        .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

040106                  PRIERR:

                        ;CLEAR USER'S ERROR CALL
040106  062716  000004          ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
040112  105076  000000          CLRB    @(SP)           ;CLEAR ERROR NUMBER
040116  162716  000004          SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

                        ;REPORT AN ERROR IF THE WRONG UNIT IS SELECTED
040122  013737  001344  001142  MOV     RMCS21,$BDDAT   ;CORRECT UNIT SELECTED??
040130  042737  177770  001142  BIC     #^CUNTMSK,$BDDAT
040136  013737  001234  001140  MOV     $UNIT,$GDDAT    ;GOOD DATA FOR TYPEOUT
040144  042737  177770  001140  BIC     #^CUNTMSK,$GDDAT
040152  123737  001140  001142  CMPB    $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                                        ;DRIVE NUMBERS
040160  001415                  BEQ     1$              ;YES!!
040162  062716  000004          ADD     #4,(SP)
040166  112776  000001  000000  MOVB    #1,@(SP)        ;ERROR 1
```

```
 58 040174  162716  000002                    SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
 59 040200  004736                            JSR     PC,@(SP)+        ;REPORT WRONG UNIT SELECTED
 60 040202  162716  000010                    SUB     #10,(SP)         ;RESTORE (SP)
 61 040206  000240                            NOP
 62 040210  000137  040730                    JMP     10$              ;SKIP OTHER CHECKS
 63 040214                          1$:
 64
 65                                 ;REPORT AN ERROR IF THE DEVICE IS NOT AVAILABLE OR IF
 66                                 ;THE DEVICE IS NONEXISTANT
 67 040214  032737  004000  001334            BIT     #DVA,RMCS11      ;DEVICE AVAILABLE??
 68 040222  001045                            BNE     5$               ;YES!!
 69 040224  013737  001334  001140            MOV     RMCS11,$GDDAT    ;EXPECTED STATUS
 70 040232  052737  004000  001140            BIS     #DVA,$GDDAT
 71 040240  013737  001334  001142            MOV     RMCS11,$BDDAT    ;RECEIVED STATUS
 72 040246  062716  000004                    ADD     #4,(SP)
 73 040252  112776  000002  000000            MOVB    #2,@(SP)         ;ERROR #2
 74 040260  032737  010000  001344            BIT     #NED,RMCS21      ;WAS NED SET??
 75 040266  001414                            BEQ     2$               ;NO!!
 76 040270  013737  001344  001140            MOV     RMCS21,$GDDAT    ;EXPECTED STATUS
 77 040276  013737  001344  001142            MOV     RMCS21,$BDDAT    ;RECEIVED STATUS
 78 040304  042737  010000  001140            BIC     #NED,$GDDAT
 79 040312  112776  000003  000000            MOVB    #3,@(SP)         ;YES - CHANGE ERROR NUMBER
 80 040320  162716  000002          2$:       SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
 81 040324  004736                            JSR     PC,@(SP)+        ;REPORT DEVICE NOT AVAILABLE
 82 040326  162716  000010                    SUB     #10,(SP)         ;RESTORE (SP)
 83 040332  000240                            NOP
 84 040334  000575                            BR      10$              ;SKIP OTHER CHECKS
 85 040336                          5$:
 86
 87                                 ;REPORT AN ERROR IF MASSBUS CONTROLLER IS NOT READY
 88 040336  032737  000200  001334            BIT     #RDY,RMCS11      ;CONTROLLER READY??
 89 040344  001030                            BNE     7$               ;YES!!
 90 040346  013737  001334  001140            MOV     RMCS11,$GDDAT    ;EXPECTED STATUS
 91 040354  052737  000200  001140            BIS     #RDY,$GDDAT
 92 040362  042737  160001  001140            BIC     #SC!TRE!MCPE!GO,$GDDAT
 93 040370  013737  001334  001142            MOV     RMCS11,$BDDAT    ;RECEIVED STATUS
 94 040376  062716  000004                    ADD     #4,(SP)
 95 040402  112776  000004  000000            MOVB    #4,@(SP)         ;ERROR #4
 96 040410  162716  000002                    SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
 97 040414  004736                            JSR     PC,@(SP)+        ;REPORT CONTROLLER NOT READY
 98 040416  162716  000010                    SUB     #10,(SP)         ;RESTORE (SP)
 99 040422  000240                            NOP
100 040424  000541                            BR      10$              ;SKIP OTHER CHECKS
101 040426                          7$:
102
103                                 ;REPORT AN ERROR IF GO IS NOT ZERO AND DRY IS NOT ONE
104 040426  032737  000001  001334            BIT     #GO,RMCS11       ;GO RESET??
105 040434  001431                            BEQ     8$               ;YES!!
106 040436  032737  000200  001346            BIT     #DRY,RMDS1       ;DRIVE READY??
107 040444  001025                            BNE     8$               ;YES!!
108 040446  013737  001334  001140            MOV     RMCS11,$GDDAT    ;EXPECTED STATUS
109 040454  042737  160001  001140            BIC     #SC!TRE!MCPE!GO,$GDDAT
110 040462  013737  001334  001142            MOV     RMCS11,$BDDAT    ;RECEIVED STATUS
111 040470  062716  000004                    ADD     #4,(SP)
112 040474  112776  000005  000000            MOVB    #5,@(SP)         ;ERROR #5
113 040502  162716  000002                    SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
114 040506  004736                            JSR     PC,@(SP)+        ;REPORT DRIVE NOT READY
```

```
                                                ;RESTORE (SP)
040510  162716  000010                SUB     #10,(SP)
040514  000240                        NOP
040516  000504                        BR      10$
040520                          8$:
                                ;REPORT AN ERROR IF THE RH CONTROLLER DETECTED BAD
                                ;PARITY ON THE MASSBUS CONTROL BUS
040520  032737  020000  001334        BIT     #MCPE,RMCS1I    ;PARITY ERROR ??
040526  001425                        BEQ     9$              ;NO!!
040530  013737  001334  001140        MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
040536  042737  160001  001140        BIC     #SC!TRE!MCPE!GO,$GDDAT
040544  013737  001334  001142        MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
040552  062716  000004                ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
040556  112776  000013  000000        MOVB    #13,@(SP)       ;ERROR #13
040564  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
040570  004736                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
040572  162716  000010                SUB     #10,(SP)        ;RESTORE STACK
040576  000240                        NOP
040600  000453                        BR      10$
040602                          9$:
                                ;REPORT AN ERROR IF DETECTED A CONTROL BUS PARITY ERROR
040602  032737  000010  001350        BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
040610  001451                        BEQ     11$             ;NO!!
040612  032737  000010  001376        BIT     #DPE,RMER2I     ;WAS IT THE CONTROL BUS??
040620  001045                        BNE     11$             ;NOT SURE!!
040622  032737  000010  001424        BIT     #PAR,RMER1O     ;DID TEST SET PAR ??
040630  001413                        BEQ     9$              ;NO!!
040632  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040634  012700  001551                MOV     #PUTINX,R0      ;R0 POINTS TO INDEX TABLE
040640  122710  000014          91$:  CMPB    #RMER1,(R0)     ;SEARCH TABLE FOR RMER1
040644  001002                        BNE     92$
040646  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
040650  000431                        BR      11$             ;PAR WAS SET BY TEST
040652  105720                  92$:  TSTB    (R0)+           ;END OF TABLE??
040654  100371                        BPL     91$             ;NO!!
040656  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
040660  013737  001350  001140  93$:  MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
040666  042737  000010  001140        BIC     #PAR,$GDDAT
040674  013737  001350  001142        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
040702  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
040706  112776  000050  000000        MOVB    #50,@(SP)       ;WRITE THE ERROR NUMBER
040714  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
040720  004736                        JSR     PC,@(SP)+       ;REPORT THE ERROR
040722  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
040726  000240                        NOP
040730  062716  000010          10$:  ADD     #10,(SP)        ;RETURN TO ERROR
040734  000240                  11$:  NOP                     ;RETURN TO NO ERROR
040736  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.
                                
040740                          SECERR:
                                
                                ;;*****************************************************************
                                ;STORE FUNCTION CODE AND CLEAR USER'S ERROR NUMBER
040740  013737  001410  044574          MOV     RMCS10,$15$     ;STORE FUNCTION CODE
040746  042737  177701  044574          BIC     #^C<F0!F1!F2!F3!F4>,$15$
040754  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
040760  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
040764  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.F., IF DRY = 0
040770  032737  000200  001346          BIT     #DRY,RMDSI      ;DRIVE READY??
040776  001024                          BNE     5$              ;YES!!
041000  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
041006  042737  177577  001142          BIC     #^CDRY,$BDDAT
041014  012737  000200  001140          MOV     #DRY,$GDDAT     ;GOOD DATA FOR TYPEOUT
041022  062716  000004                  ADD     #4,(SP)
041026  112776  000010  000000          MOVB    #10,@(SP)       ;ERROR NUMBER
041034  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
041040  004736                          JSR     PC,@(SP)+       ;REPORT NOT READY
041042  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP) TO ERROR N
041046  000240                          NOP
                                
                                  EPORT ERROR IF GO BIT IS NOT RESET
041050  032737  000001  001334  5$:     BIT     #GO,RMCS11      ;GO BIT RESET??
041056  001423                          BEQ     10$             ;YES!!
041060  013737  001334  001142          MOV     RMCS1i,$BDDAT   ;BAD DATA FOR TYPEOUT
041066  042737  177776  001142          BIC     #^CGO,$BDDAT
041074  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
041100  062716  000004                  ADD     #4,(SP)
041104  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$:    MOV     RMCS11,$BDDAT   ;IS FUNCTION CODE CORRECT??
                                        BIC     #^C76,$BDDAT
                                        MOV     515$,$GDDAT     ;EXPECTED FUNCTION CODE
                                        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     $GDDAT          ,EXPECT "ERR" = 0
                                        TST     RMER11          ;IS RMER1 = 0??
                                        BNE     20$             ;NO!!
                                        TST     RMER21          ;IS RMER2 = 0??
                                        BEQ     25$             ;YES!!
                                20$:    BIS     #ERR,$GDDAT     ;"ERR" SOULD BE SET
                                25$:    MOV     RMDS1,$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     $GDDAT          ;EXPECT "TRE" = 0
                                        MOV     RMCS21,-(SP)    ;WAS DLT, WCE, UPE, NED, NEM
                                        BIC     #377,(SP)+      ;PGE, MXF OR MDPE SET
                                        BNE     35$             ;YES!!
                                        BIT     #ERR,RMDS1      ;WAS EXCEPTION RECEIVED??
                                        BEQ     40$             ;NO!!
                                        CMP     #SEARCH,515$    ;WAS DATA TRANSFERRED??
                                        BHIS    40$             ;NO!!
                                35$:    BIS     #TRE,$GDDAT     ;"TRE" SHOULD BE SET
                                40$:    MOV     RMCS11,$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 CAL.
                                        MOVB    #14,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT TRE ERROR
```

```
041414  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
041420  000240                          NOP
041422                          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 ENTRY FROM THE FUNCTION CODE TABLE
041422  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
041424  013700  044574                  MOV     515$,R0         ;GET FUNCTION CODE
041430  016037  064406  044566          MOV     FNCDTB(R0),500$ ;STORE ENTRY
041436  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.
041440  013737  044566  001140          MOV     500$,$GDDAT     ;GET EXPECTED ATA STATUS
041446  032737  040000  001346          BIT     #ERR,RMDSI      ;IS COMPOSITE ERROR SET ??
041454  001403                          BEQ     50$             ;NO !!
041456  052737  100000  001140          BIS     #ATA,$GDDAT     ;EXPECT AN ATTENTION
041464  042737  077777  001140  50$:    BIC     #^CATA,$GDDAT   ;STRIP DONT CARES
041472  013737  001346  001142          MOV     RMDSI,$BDDAT    ;GET RECEIVED ATA
041500  042737  077777  001142          BIC     #^CATA,$BDDAT   ;STRIP DONT CARES
041506  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA OK ??
041514  001413                          BEQ     55$             ;YES !!
041516  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
041522  112776  000006  000000          MOVB    #6,@(SP)        ;LOAD ERROR # IN CALL
041530  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
041534  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
041536  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
041542  000240                          NOP
041544                          55$:

                                ;REPORT ERROR IF ILF IS INCORRECT, I.E., IF ILF DOES NOT COMPARE
                                ;WITH FUNCTION CODE TABLE
041544  013737  044566  001140          MOV     500$,$GDDAT     ;GET EXPECTED ILF
041552  042737  177776  001140          BIC     #^CILF,$GDDAT   ;CLEAR ALL OTHER BITS
041560  013737  001350  001142          MOV     RMER11,$BDDAT   ;GET RECEIVED ILF
041566  042737  177776  001142          BIC     #^CILF,$BDDAT   ;CLEAR ALL OTHER BITS
041574  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ILF OK ??
041602  001412                          BEQ     60$             ;YES !!
041604  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
041610  112776  000254  000000          MOVB    #254,@(SP)      ;WRITE ERROR NUMBER IN CALL
041616  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
041622  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
041624  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
041630  005037  001140          60$:    CLR     $GDDAT          ;CLEAR EXPECTED STATUS

                                ;REPORT AN ERROR IF WCE IS SET AND SHOULD NOT BE SET
041634  013746  044566                  MOV     500$,-(SP)      ;GET WCE STATUS ENABLE
041640  052716  137777                  BIS     #^CWCE,(SP)     ;SET ALL OTHER BITS
041644  013737  001344  001142          MOV     RMCS21,$BDDAT   ;RECEIVED STATUS
041652  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
```

```
    229 042104  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    230 042110  112776  000023  000000              MOVB    #23,@(SP)       ;WRITE ERROR NUMBER IN CALL
    231 042116  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    232 042122  004736                              JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    233 042124  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    234 042130                              125$:
    235
    236                                     ;REPORT ERROR IF WCF IS SET AND WRITE ERRORS ARE NOT ENABLED
    237 042130  012746  177777                      MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
    238 042134  032737  004000  044566              BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
    239 042142  001002                              BNE     130$            ;YES !!
    240 042144  042716  000040                      BIC     #WCF,(SP)       ;DISABLE WCF ERROR
    241 042150  013737  001350  001142  130$:       MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
    242 042156  042637  001142                      BIC     (SP)+,$BDDAT    ;RESET WCF IF ENABLED
    243 042162  001412                              BEQ     135$            ;BRANCH IF WCF OK
    244 042164  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    245 042170  112776  000025  000000              MOVB    #25,@(SP)       ;WRITE ERROR NUMBER IN CALL
    246 042176  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    247 042202  004736                              JSR     PC,@(SP)+       ;REPORT ERROR
    248 042204  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    249 042210                              135$:
    250
    251                                     ;REPORT ERROR IF DPE IS SET AND WRITE ERRORS ARE NOT ENABLED
    252 042210  012746  177777                      MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
    253 042214  032737  004000  044566              BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
    254 042222  001002                              BNE     140$            ;YES !!
    255 042224  042716  000010                      BIC     #DPE,(SP)       ;RESET DPE ENABLE
    256 042230  013737  001376  001142  140$:       MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
    257 042236  042637  001142                      BIC     (SP)+,$BDDAT    ;RESET DPE IF ENABLED
    258 042242  001412                              BEQ     145$            ;BRANCH IF DPE OK
    259 042244  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    260 042250  112776  000040  000000              MOVB    #40,@(SP)       ;WRITE ERROR NUMBER IN CALL
    261 042256  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    262 042262  004736                              JSR     PC,@(SP)+       ;REPORT ERROR
    263 042264  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    264 042270                              145$:
    265
    266                                     ;REPORT AN ERROR IF UPE IS SET AND WRITE ERRORS ARE NOT ENABLED
    267 042270  012746  177777                      MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
    268 042274  032737  004000  044566              BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
    269 042302  001002                              BNE     150$            ;YES !!
    270 042304  042716  020000                      BIC     #UPE,(SP)       ;DISABLE UPE ERROR
    271 042310  013737  001344  001142  150$:       MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
    272 042316  042637  001142                      BIC     (SP)+,$BDDAT    ;RESET UPE IF ENABLED
    273 042322  001412                              BEQ     155$            ;BRANCH IF UPE OK
    274 042324  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    275 042330  112776  000024  000000              MOVB    #24,@(SP)       ;WRITE ERROR NUMBER IN CALL
    276 042336  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    277 042342  004736                              JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    278 042344  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    279 042350                              155$:
    280
    281                                     ;REPORT AN ERROR IF IAE IS SET AND IS NOT ENABLED
    282 042350  013746  044566                      MOV     500$,-(SP)      ;GET IAE ENABLE
    283 042354  052716  175777                      BIS     #^C1AE,(SP)     ;SET ALL OTHER BITS
    284 042360  013737  001350  001142              MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
    285 042366  042637  001142                      BIC     (SP)+,$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 !!
```

```
                                                ;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
042730  032737  002000  001366                  BIT     #HCI,RMOFI      ;IS HCI SET ??
042736  001403                                  BEQ     201$            ;NO !!
042740  042737  000200  044566                  BIC     #HCE,500$       ;YES - DISABLE ALL HEADER ERRORS
042746                                  201$:
                                                ;REPORT AN ERROR IF HCRC IS SET AND HEADER ERRORS ARE NOT ENABLED
042746  012746  177777                          MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
042752  032737  000200  044566                  BIT     #HCE,500$       ;ARE HEADER ERRORS ENABLED ??
042760  001002                                  BNE     205$            ;YES !!
042762  042716  000400                          BIC     #HCRC,(SP)      ;DISABLE HCRC
042766  013737  001350  001142          205$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
042774  042637  001142                          BIC     (SP)+,$BDDAT    ;RESET HCRC IF ENABLED
043000  001412                                  BEQ     210$            ;BRANCH IF HCRC IS OK
043002  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
043006  112776  000035  000000                  MOVB    #35,@(SP)       ;WRITE ERROR NUMBER IN CALL
043014  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
043020  004736                                  JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
043022  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR
043026                                  210$:

                                                ;REPORT ERROR IF HCE IS SET AND HEADER ERRORS ARE NOT ENABLED
043026  012746  177777                          MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
043032  032737  000200  044566                  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
044566                                          BIT     #HCE,$OO$       ;ARE HEADER ERRORS 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
044566                                          BIT     #HCE,$OO$       ;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 OF 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
044566                                          BIT     #ECH,$OO$       ;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
```

```
                                                BEQ     250$            ;BRANCH IF MDPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #27,@(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
                                        250$:

                                        ;REPORT ERROR IF DCK IS SET AND DATA FIELD ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                                BIT     #ECH,500$       ;ARE THEY ENABLED ??
                                                BNE     255$            ;YES !!
                                                BIC     #DCK,(SP)       ;DISABLE DCK
                                        255$:   MOV     RMER11,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR DCK IF ENABLED
                                                BEQ     260$            ;BRANCH IF DCK IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #30,@(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
                                        260$:

                                        ;REPORT ERROR IF ECH IS SET AND,
                                        ;       DATA FIELD ERRORS ARE NOT ENABLED, OR
                                        ;       ECI IS SET, OR
                                        ;       DCK IS NOT SET.
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                                BIT     #ECH,500$       ;ARE ERRORS ENABLED ??
                                                BEQ     265$            ;NO !!
                                                BIT     #ECI,RMOF1      ;IS ECI SET ??
                                                BNE     265$            ;YES !!
                                                BIT     #DCK,RMER11     ;IS DCK ALSO SET ??
                                                BNE     270$            ;YES !!
                                        265$:   BIC     #ECH,(SP)       ;DISABLE ECH
                                        270$:   MOV     RMER11,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR ECH IF ENABLED
                                                BEQ     275$            ;BRANCH IF ECH IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #31,@(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
                                        275$:
```

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

043506  022737  000030  044574            CMP     #SEARCH,$15$    ;WAS DATA TRANSFERRED ?
043514  103402                            BLO     280$            ;BR IF YES
043516  000137  044540                    JMP     355$            ;NO - EXIT

                                    ;REPORT ERROR IF RMWC NOT ZERO AND TRE IS ZERO
043522  013737  001336  001142    280$:   MOV     RMWC1,$BDDAT    ;WORD COUNT ZERO??
043530  001421                            BEQ     285$            ;YES
043532  032737  040000  001334            BIT     #TRE,RMCS11     ;TRANSFER ERROR DETECTED??
043540  001015                            BNE     285$            ;YES!!
043542  062716  000004                    ADD     #4,(SP)
043546  112776  000015  000000            MOVB    #15,@(SP)       ;ERROR NUMBER
043554  005037  001140                    CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
043560  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
043564  004736                            JSR     PC,@(SP)+       ;REPORT WORD COUNT NOT ZERO
043566  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
043572  000240                            NOP

                                    ;REPORT ERROR IF RMBA IS NOT CORRECT
043574  013737  001336  001140    285$:   MOV     RMWC1,$GDDAT    ;GET WORD COUNT AT END OF TRANSFER AND
043602  163737  001412  001140            SUB     RMWC0,$GDDAT    ;SUBTRACT STARTING WORD COUNT.
043610  006337  001140                    ASL     $GDDAT          ;* 2
043614  063737  001414  001140            ADD     RMBA0,$GDDAT    ;ADD STARTING BUS ADDRESS

043622  032737  000010  001344            BIT     #BAI,RMCS1     ;WAS BUS ADDRESS INHIBIT (BAI) SET ??
043630  001403                            BEQ     290$            ;NO !!
043632  013737  001414  001140            MOV     RMBA0,$GDDAT    ;ADDRESS SHOULD NOT HAVE CHANGED

043640  023737  001140  001340    290$:   CMP     $GDDAT,RMBAI    ;BUS ADDRESS OK??
043646  001416                            BEQ     295$            ;YES!!
043650  013737  001340  001142            MOV     RMBA1,$BDDAT    ;BAD DATA FOR TYPEOUT
043656  062716  000004                    ADD     #4,(SP)
043662  112776  000016  000000            MOVB    #16,@(SP)       ;ERROR NUMBER
043670  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
043674  004736                            JSR     PC,@(SP)+       ;REPORT UNEXPECTED ADDRESS
043676  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
043702  000240                            NOP

                                    ;COMPUTE NUMBER OF SECTORS TRANSFERRED FROM WORD COUNT
043704  005046                    295$:   CLR     -(SP)           ;NUMBER OF SECTORS TRANSFERRED
043706  013746  001336                    MOV     RMWC1,-(SP)     ;GET WORD COUNT AT END OF TRANSFER AND
043712  163716  001412                    SUB     RMWC0,(SP)      ;SUBTRACT STARTING WORD COUNT.

043716  012746  000400                    MOV     #256.,-(SP)     ;ASSUME 256. WORDS PER SECTOR
043722  032737  000002  001410            BIT     #BIT1,RMCS10    ;HEADER & DATA COMMAND ??
043730  001402                            BEQ     300$            ;NO !!
043732  062716  000002                    ADD     #2,(SP)         ;CHANGE TO 258. WORDS PER SECTOR

043736  005266  000004            300$:   INC     4(SP)           ;INCREMENT SECTOR COUNT
043742  161666  000002                    SUB     (SP),2(SP)      ;SUBTRACT ONE SECTOR'S WORTH
043746  003373                            BGT     300$            ;CONTINUE IF NOT DONE
043750  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK

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

```
                                        ;NUMBER OF SECTORS
043752  013737  001444  044566        MOV     RMDC0,500$      ;STORE ORIGINAL CYLINDER
043760  013737  001416  044570        MOV     RMDA0,505$      ;STORE ORIGINAL TRACK
043766  013737  001416  044572        MOV     RMDA0,510$      ;STORE ORIGINAL SECTOR
043774  013737  001332  044576        MOV     LSTRK,520$      ;STORE LAST TRACK,
044002  000337  044576                SWAB    520$            ;GET TRACK ADDRESS TO LO BYTE AND
044006  005237  044576                INC     520$            ;INCREMENT TO GET TOTL # OF TRACKS.

044012  042737  000377  044570        BIC     #^C<TADMSK>,505$        ;SAVE TRACK ADDRESS BITS AND
044020  000337  044570                SWAB    505$            ;SWAP TRACK ADDRESS TO LOW BYTE.
044024  042737  177400  044572        BIC     #^C<SADMSK>,510$        ;SAVE SECTOR ADDRESS BITS
044032  062637  044572                ADD     (SP)+,510$

044036  023727  044572  000040  310$: CMP     510$,#32.       ;SECTOR OVEFLOWED??
044044  103406                        BLO     315$            ;NO!!
044046  005237  044570                INC     505$            ;INCREMENT TRACK
044052  162737  000040  044572        SUB     #32.,510$       ;ADJUST SECTOR
044060  000766                        BR      310$            ;TRY AGAIN

044062  023737  044570  044576  315$: CMP     505$,520$       ;TRACK OVERFLOWED??
044070  10? 07                        BLO     320$            ;NO!!
044072  005237  044566                INC     500$            ;INCREMENT CYLINDER
044076  163737  044576  044570        SUB     520$,505$       ;AD UST TRACK
044104  000766                        BR      315$            ;TRY AGAIN
044106  000240                        NOP

                                        ;REPORT ERROR IF "LBT" IS NOT CORRECT
044110                          320$:
044110  005037  001140                CLR     $GDDAT          ;SET GOOD DATA FOR LBT = 0
044114  023727  044566  001466        CMP     500$,#822.      ;SHOULD LBT BE SET??
044122  101407                        BLOS    325$            ;NO!!
044124  032737  002000  001350        BIT     #IAE,RMER1I     ;WAS IAE SET ??
044132  001003                        BNE     325$            ;YES - LBT SHOULD NOT BE SET
044134  012737  002000  001140        MOV     #LBT,$GDDAT     ;SET GOOD DATA FOR LBT = 1
044142  013737  001346  001142  325$: MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
044150  042737  175777  001142        BIC     #^CLBT,$BDDAT
044156  023737  001140  001142        CMP     $GDDAT,$BDDAT   ;IS LBT CORRECT??
044164  001413                        BEQ     330$            ;YES!!
044166  062716  000004                ADD     #4,(SP)
044172  112776  000017  000000        MOVB    #17,@(SP)       ;ERROR NUMBER
044200  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
044204  004736                        JSR     PC,@(SP)+       ;REPORT LBT IS WRONG
044206  162716  000010                SUB     #10,(SP)        ;RESTORE (SP)
044212  000240                        NOP

                                        ;REPORT ERROR IF "AOE" IS INCORRECT
044214  005037  001140          330$: CLR     $GDDAT          ;SET FOR AOE = 0
044220  032737  002000  001350        BIT     #IAE,RMER1I     ;WAS "IAE" DETECTED??
044226  001031                        BNE     340$            ;YES-"AOE" SHOULD BE ZERO
044230  023727  044566  001466        CMP     500$,#822.      ;SHOULD AOE BE SET??
044236  101425                        BLOS    340$            ;NO!!
044240  005737  044570                TST     505$            ;MAYBE
044244  001012                        BNE     335$            ;YES
044246  005737  044572                TST     510$
044252  001007                        BNE     335$            ;YES !!
044254  032737  000010  044574        BIT     #F2,515$        ;WAS THIS READ OR WRITE CHECK ??
044262  001413                        BEQ     340$            ;NO !!
```

```
                                                TST     RMWCI           ;WAS ALL DATA TRANSFERRED ??
                                                BEQ     340$            ;YES !!
                                        335$:   MOV     #AOE,$GDDAT     ;SET FOR AOE = 1
                                                CLR     505$            ;CLEAR EXPECTED TRACK
                                                MOV     #1,510$         ;EXPECT SECTOR = 1
                                        340$:   MOV     RMER1I,$BDDAT   ;BAD DATA FOR TYPEOUT
                                                BIC     #^CAOE,$BDDAT
                                                CMP     $GDDAT,$BDDAT   ;IS AOE CORRECTY??
                                                BEQ     345$            ;YES!!
                                                ADD     #4,(SP)
                                                MOVB    #20,@(SP)       ;ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT AOE IS WRONG
                                                SUB     #10,(SP)        ;RESTORE (SP)
                                                NOP

                                        ;REPORT ERROR IF RMDA IS NOT CORRECT
                                        345$:   BIT     #IAE,RMER1I     ;WAS THERE AN IAE ERROR ??
                                                BNE     355$            ;YES - DONT CHECK RMDA,RMDC
                                                MOV     505$,$GDDAT     ;SETUP EXPECTED DISK ADDRESS
                                                SWAB    $GDDAT
                                                MOVB    510$,$GDDAT
                                                MOV     RMDAI,$BDDAT    ;SETUP RECEIVED DISK ADDRESS
                                                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
                                                BEQ     350$            ;BRANCH IF EQUAL
                                                ADD     #4,(SP)
                                                MOVB    #21,@(SP)       ;ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT BAD DISK ADDRESS
                                                SUB     #10,(SP)        ;RESTORE (SP)
                                                NOP

                                        ;REPORT ERROR IF RMDC IS INCORRECT
                                        350$:   MOV     500$,$GDDAT     ;SETUP EXPECTED CYLINDER
                                                BIC     #^C1777,$GDDAT
                                                MOV     RMDCI,$BDDAT    ;SETUP RECEIVED CYLINDER
                                                CMP     $GDDAT,$BDDAT   ;COMPAPE CYLINDERS
                                                BEQ     355$            ;BRANCH IF EQUAL
                                                ADD     #4,(SP)
                                                MOVB    #22,@(SP)       ;ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT BAD CYLINDER
                                                SUB     #10,(SP)        ;RESTORE (SP)
                                                NOP

                                        355$:   ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
                                                TSTB    @(SP)           ;WAS ERROR FOUND??
                                                BEQ     360$
                                                ADD     #4,(SP)         ;MOVE (SP) TO ERROR RETURN
                                                BR      365$
                                        360$:   SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
                                        365$:   RTS     PC

                                        500$:   .WORD   0               ;CYLINDER
                                        505$:   .WORD   0               ;TRACK
                                        510$:   .WORD   0               ;SECTOR
                                        515$:   .WORD   0               ;FUNCTION CODE
```

```
    172 044576  000000                520$:   .WORD   0               ;# OF TRACKS FOR DEVICE UNDER TEST = LAST
    173                                                               ;TRACK + 1 TRACK
    174
```

```
                                .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

044600                          CMPERRSTS:

                                ;MASK AND STORE THE CONTENTS OF RMER1  AND RMER2
044600  013737  001350  001176          MOV     RMER1I,$TMP1    ;STORE RMER1 AT TEMP STORAGE
044606  047637  000000  001176          BIC     @(SP),$TMP1     ;MASK   RMER1
044614  062716  000002                  ADD     #2,(SP)         ;MOVE SP TO NEXT MASK
044620  013737  001376  001200          MOV     RMER2I,$TMP2    ;STORE RMER2 AT TEMP STORAGE
044626  047637  000000  001200          BIC     @(SP),$TMP2     ;MASK RMER2

                                ;CLEAR USER'S ERROR CALL
044634  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USER'S ERROR CALL
044640  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
044644  162716  000004                  SUB     #4,(SP)         ;LEAVE SP AT NO ERROR RETURN

                                ;SEE IF THERE WERE ANY ERRORS IN RMER1, I.E., $TMP1
044650  005737  001176                  TST     $TMP1           ;ANY ERRORS TO REPORT??
044654  001420                          BEQ     5$              ;NO !!
044656  013737  001176  001142          MOV     $TMP1,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
044664  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
044670  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
044674  112776  000066  000000          MOVB    #66,@(SP)       ;CORRECTABLE DATA CHECK ERROR #
044702  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
044706  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
044710  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
044714  000240                          NOP
044716                          5$:

                                ;SEE IF THERE ARE ANY ERRORS TO REPORT IN RMER2 ($TMP2)
044716  005737  001200                  TST     $TMP2           ;ANY ERRORS IN RMER2?
044722  001420                          BEQ     10$             ;NO!!

044724  013737  001200  001142          MOV     $TMP2,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
044732  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
044736  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
044742  112776  000067  000000          MOVB    #67,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
044750  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
```

```
      58 044754  004736                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
      59 044756  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
      60 044762  000240                        NOP
      61 044764                          10$:
      62
      63                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
      64 044764  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
      65 044770  105776  000000                TSTB    @(SP)           ;WAS THERE AN ERROR CALLED??
      66 044774  001403                        BEQ     20$             ;NO!!
      67 044776  062716  000004                ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
      68 045002  000402                        BR      30$
      69 045004  162716  000004          20$:  SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
      70 045010  000207                  30$:  RTS     PC              ;RETURN TO USER
      71
```

```
                                        .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

045012                                  DEVSEL:

                                        ;CLEAR USER'S ERROR CALL
045012  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
045016  105076  000000                          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF CALL
045022  162716  000004                          SUB     #4,(SP)         ;MOVE SP BACK

                                        ;SAVE USER'S INFORMATION AND SETUP REGISTERS
045026  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
045032  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
045036  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
045040  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
045042  012737  045162  000004                  MOV     #20$,ERRVEC     ;SETUP FOR BUS TIMEOUT
045050  012737  000300  000006                  MOV     #PR6,ERRVEC+2
045056  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
045062  013701  001464                          MOV     TSTQUE,R1       ;R1 POINTS TO DEVICE NUMBER

                                        ;SELECT DEVICE AND VERIFY THAT DEVICE IS AVAILABLE
045066  111160  000010                          MOVB    (R1),RMCS2(R0)  ;WRITE UNIT SELECT BITS
045072  016037  000000  001176                  MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
045100  016037  000010  001174                  MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS

045106  032737  010000  001174                  BIT     #NED,$TMP0      ;IS DEVICE NONEXISTENT ?
045114  001407                                  BEQ     10$             ;NO!!
045116  062766  000004  000010                  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
045124  112776  000111  000010                  MOVB    #111,@10(SP)    ;WRITE ERROR NUMBER
045132  000422                                  BR      30$

045134  032737  004000  001176  10$:            BIT     #DVA,$TMP1      ;IS DEVICE AVAILABLE ?
045142  001021                                  BNE     35$             ;YES!!
045144  062766  000004  000010                  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
045152  112776  000112  000010                  MOVB    #112,@10(SP)    ;WRITE ERROR NUMBER
045160  000407                                  BR      30$

                                        ;HANDLE BUS TIMEOUT
045162  022626                          20$:    CMP     (SP)+,(SP)+     ;ADJUST SP
045164  062766  000004  000010                  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
045172  112776  000113  000010                  MOVB    #113,@10(SP)    ;WRITE BUS TIMEOUT ERROR NUMBER
045200  162766  000002  000010  30$:            SUB     #2,10(SP)       ;ADJUST RETURN TO 'NOP' PRECEDING
                                                                        ;THE ERROR CALL

                                        ;RESTORE USERS DATA AND RETURN TO ADDRESS ON STACK
045206                                  35$:
045206  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
045210  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
045212  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
```

SEQ 0192

```
        045216  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     52 045222  000207                          RTS     PC              ;EXIT
     53
```

```
                                .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

045224                          SEKSTS:

                                ;CLEAR USERS' ERROR CALL
                                        NOP
045224  000240                         ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
045226  062716  000004                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
045232  105076  000000                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
045236  162716  000004                  CLR     300$            ;CLEAR ERROR FLAGS
045242  005037  046462

                                ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN WRITING
                                ;LOCAL REGISTERS, I.E., "PAR"  = 1 AND "DPE" = 0
045246  032737  000010  001350          BIT     #PAR,RMER1I     ;WAS PARITY ERROR DETECTED??
045254  001424                          BEQ     1$              ;NO!!
045256  032737  000010  001376          BIT     #DPE,RMER2I     ;WAS IT DUE TO CONTROL BUS??
045264  001020                          BNE     1$              ;NOT SURE!!

                                ;REPORT REGISTER PARITY ERROR VIA USER'S ERROR CALL
045266  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
045272  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
045300  062716  000004                  ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
045304  112776  000050  000000          MOVB    #50,@(SP)       ;ERROR #50
045312  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045316  004736                          JSR     PC,@(SP)+
045320  162716  000010                  SUB     #10,(SP)        ;RESTORE STACK
045324  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.
045326  012737  002000  001140  1$:     MOV     #IAE,$GDDAT     ;SETUP FOR IAE = 1
045334  052737  040000  046462          BIS     #SKI,300$       ;SETUP FOR SKI = 1
045342  023727  001444  001466          CMP     RMDC0,#822.     ;GREATER THAN LAST CYLINDER ?
045350  101025                          BHI     3$              ;YES - CYLINDER IS INVALID
045352  042737  040000  046462          BIC     #SKI,300$       ;CLEAR SKI ERROR FLAG

045360  123737  001417  001333          CMPB    RMDA0+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
045366  101016                          BHI     3$              ;YES - TRACK IS INVALID

045370  123727  001416  000035          CMPB    RMDA0,#29.      ;SECTOR > 29. ?
045376  101410                          BLOS    2$              ;BR IF NO
```

```
     045400  032737  010000  001442          BIT     #FMT16,RMOFO    ;18 BIT FORMAT ?
     045406  001406                          BEQ     3$              ;YES - SECTOR IS INVALID FOR 18 BIT MODE
     045410  123727  001416  000037          CMPB    RMDAO,#31.      ;SECTOR > 31. ?
     045416  101002                          BHI     3$              ;YES - SECTOR IS INVALID

     045420  005037  001140          2$:     CLR     $GDDAT          ;"IAE" SHOULD = 0

                                     ;COMPARE EXPECTED AND RECIEVED "IAE" STATUS
     045424  013737  001350  001142 3$:     MOV     RMER1I,$BDDAT   ;IS IAE OK??
     045432  042737  175777  001142         BIC     #^CIAE,$BDDAT   ;SAVE IAE BIT FOR COMPARE
     045440  023737  001140  001142         CMP     $GDDAT,$BDDAT   ;CORRECT "IAE" STATUS ?
     045446  001004                          BNE     35$             ;BR IF NO
     045450  042737  040000  046462         BIC     #SKI,300$       ;CLEAR SKI FLAG
     045456  000413                          BR      5$              ;GO CHECK NEXT ERROR
     045460                          35$:
                                     ;REPORT INCORRECT "IAE" STATUS VIA USER'S ERROR CALL
     045460  062716  000004                  ADD     #4,(SP)
     045464  112776  000051  000000         MOVB    #51,@(SP)       ;ERROR 51
     045472  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     045476  004736                          JSR     PC,@(SP)+       ;REPORT INCORRECT IAE
     045500  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP)
     045504  000240                          NOP
     045506                          5$:

                                     ;REPORT ANY IVC ERROR AS
                                     ;       IVC ERROR WITH VOLUME VALID ZERO
                                     ;       ERRONEOUS IVC ERROR, VOLUME VALID IS SET
     045506  032737  010000  001376         BIT     #IVC,RMER2I     ;IVC ERROR??
     045514  001427                          BEQ     52$             ;NO!!
     045516  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
     045522  013737  001376  001142         MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     045530  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
     045534  112776  000060  000000         MOVB    #60,@(SP)       ;ERROR 60 IF VV = 0
     045542  032737  000100  001346         BIT     #VV,RMDSI
     045550  001403                          BEQ     51$
     045552  112776  000061  000000         MOVB    #61,@(SP)       ;ERROR 61 IF VV  = 1
     045560  162716  000002          51$:    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     045564  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     045566  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
     045572  000240                          NOP

     045574  013737  001376  001142 52$:     MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     045602  042737  137777  001142         BIC     #^CSKI,$BDDAT   ;CLEAR DONT CARES
     045610  013737  046462  001140         MOV     300$,$GDDAT     ;GET EXPECTED SKI S  TUS
     045616  042737  137777  001140         BIC     #^CSKI,$GDDAT   ;CLEAR DONT CARES
     045624  001417                          BEQ     53$             ;BRANCH IF 0 EXPECTED

                                     ;REPORT ERROR IF SKI IS NOT SET (IAE WAS NOT DETECTED)
     045626  032737  040000  001142         BIT     #SKI,$BDDAT     ;WAS SKI DETECTED ??
     045634  001032                          BNE     54$             ;YES !!
     045636  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
     045642  112776  000267  000000         MOVB    #267,@(SP)      ;WRITE ERROR NUMBER
     045650  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
     045654  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
     045656  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
     045662  000443                          BR      6$              ;GO TO NEXT ERROR CHECK
     045664                          53$:
```

```
                                        ;REPORT ERROR IF SKI IS SET
045664  032737  040000  001142          BIT     #SKI,$BDDAT     ;IS SKI SET ??
045672  001413                          BEQ     54$             ;NO - SKI IS OK
045674  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
045700  112776  000054  000000          MOVB    #54,@(SP)       ;LOAD ERROR NUMBER
045706  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045712  004736                          JSR     PC,@(SP)+       ;REPORT SEEK ERROR
045714  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
045720  000240                          NOP

                                        ;REPORT ANY DEVICE CHECK
045722  032737  000200  001376  54$:    BIT     #DVC,RMER2I     ;WAS THERE DVC DURING SEEK??
045730  001420                          BEQ     6$              ;NO!!
045732  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
045736  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
045744  062716  000004                  ADD     #4,(SP)
045750  112776  000055  000000          MOVB    #55,@(SP)       ;ERROR  #55
045756  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045762  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
045764  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
045770  000240                          NOP

                                        ;REPORT ANY "OPI" ERROR AS OPI WITH MOL  = 0, OR  OPI
                                        ;BECAUSE ON CYLINDER LATCH DIDN'T RESET
045772  032737  020000  001350  6$:     BIT     #OPI,RMER1I     ;"OPI" ERROR??
046000  001427                          BEQ     8$              ;NO!!
046002  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
046006  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
046014  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
046020  112776  000052  000000          MOVB    #52,@(SP)       ;LOAD ERROR NUMBER
046026  032737  010000  001346          BIT     #MOL,RMDSI      ;WAS MEDIUM ON LINE??
046034  001403                          BEQ     7$              ;NO!!
046036  112776  000053  000000          MOVB    #53,@(SP)       ;YES - CHANGE ERROR NUMBER
046044  162716  000002          7$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046050  004736                          JSR     PC,@(SP)+       ;REPORT "OPI" ERROR
046052  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
046056  000240                          NOP

                                        ;SEE IF "PIP" = 0, AND "ATA", "MOL" AND "VV" = 1
046060  013746  001346          8$:     MOV     RMDSI,-(SP)
046064  042716  047677                  BIC     #^C<ATA!PIP!MOL!VV>,(SP)
046070  022726  110100                  CMP     #ATA!MOL!VV,(SP)+
046074  001002                          BNE     9$              ;ERROR IN RMDS
046076  000137  046432                  JMP     14$             ;RMDS IS OK

                                        ;REPORT ERROR IF MOL = 0 AND OPI = 0
046102  032737  010000  001346  9$:     BIT     #MOL,RMDSI      ;IS MOL RESET??
046110  001030                          BNE     10$             ;NO - MOL IS SET
046112  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET
046120  001024                          BNE     10$             ;YES - DONT REPORT ERROR
046122  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
046130  052737  010000  001140          BIS     #MOL,$GDDAT
046136  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
046144  062716  000004                  ADD     #4,(SP)
046150  112776  000062  000000          MOVB    #62,@(SP)
046156  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
```

```
046162  004736                                   JSR     PC,@(SP)+          ;REPORT ERROR VIA USER
046164  162716  000010                           SUB     #10,(SP)
046170  000240                                   NOP

                                         ;REPORT AN ERROR IF "PIP" IS STIL SET AND SKI NOT SET
046172  032737  020000  001346           10$:    BIT     #PIP,RMDSI         ;IS "PIP" STILL SET??
046200  001430                                   BEQ     11$                ;NO!!
046202  032737  040000  001376                   BIT     #SKI,RMER2I        ;WAS "SKI"SET??
046210  001024                                   BNE     11$                ;YES-DONT REPORT PIP
046212  013737  001346  001140                   MOV     RMDSI,$GDDAT       ;EXPECTED STATUS
046220  042737  020000  001142                   BIC     #PIP,$BDDAT
046226  013737  001346  001142                   MOV     RMDSI,$BDDAT       ;RECEIVED STATUS
046234  062716  000004                           ADD     #4,(SP)            ;MOVE (SP) TO ERROR
046240  112776  000056  000000                   MOVB    #56,@(SP)          ;LOAD ERROR NUMBER
046246  162716  000002                           SUB     #2,(SP)            ;MOVE SP TO RETURN FOR ERROR
046252  004736                                   JSR     PC,@(SP)+          ;REPORT "PIP" SET AFTER SEEK
046254  162716  000010                           SUB     #10,(SP)                   ;RESTORE (SP)
046260  000240                                   NOP

                                         ;REPORT AN ERROR IF "ATA" IS NOT SET
046262  032737  100000  001346           11$:    BIT     #ATA,RMDSI         ;WAS "ATA" SET ??
046270  001024                                   BNE     13$                ;YES!!
046272  013737  001346  001140                   MOV     RMDSI,$GDDAT       ;EXPECTED STATUS
046300  052737  110600  001140                   BIS     #ATA!MOL!DPR!DRY,$GDDAT
046306  013737  001346  001142                   MOV     RMDSI,$BDDAT       ;RECEIVED STATUS
046314  062716  000004                           ADD     #4,(SP)            ;MOVE (SP) TO ERROR
046320  112776  000057  000000                   MOVB    #57,@(SP)          ;LOAD ERROR NUMBER
046326  162716  000002                           SUB     #2,(SP)            ;MOVE SP TO RETURN FOR ERROR
046332  004736                                   JSR     PC,@(SP)+          ;REPORT ATTENTION NOT SET DURING
                                                                            ;SEEK TEST
046334  162716  000010                           SUB     #10,(SP)                   ;RESTORE (SP)
046340  000240                                   NOP

                                         ;REPORT ERROR IF VOLUME VALID IS RESET AND IVC IS ZERO
046342  032737  000100  001346           13$:    BIT     #VV,RMDSI          ;IS VV = 0 ??
046350  001030                                   BNE     14$                ;NO!!
046352  032737  010000  001376                   BIT     #IVC,RMER2I        ;IS IVC ALSO 0 ??
046360  001024                                   BNE     14$                ;NO - IVC IS SET
046362  013737  001346  001140                   MOV     RMDSI,$GDDAT       ;EXPECTED STATUS
046370  052737  000100  001140                   BIS     #VV,$GDDAT
046376  013737  001346  001142                   MOV     RMDSI,$BDDAT       ;RECEIVED STATUS
046404  062716  000004                           ADD     #4,(SP)
046410  112776  000064  000000                   MOVB    #64,@(SP)          ;ERROR #64
046416  162716  000002                           SUB     #2,(SP)            ;MOVE SP TO RETURN FOR ERROR
046422  004736                                   JSR     PC,@(SP)+
046424  162716  000010                           SUB     #10,(SP)
046430  000240                                   NOP
046432                                   14$:

                                         ;MODIFY THE RETURN ADDRESS IF AN ERROR WAS DETECTED
046432  000240                                   NOP
046434  062716  000004                           ADD     #4,(SP)            ;MOVE (SP) TO ERROR CALL
046440  105776  000000                           TSTB    @(SP)              ;WAS ERROR CALLED??
046444  001403                                   BEQ     15$                ;NO!!
046446  062716  000004                           ADD     #4,(SP)            ;MOVE TO ERROR RETURN
046452  000402                                   BR      16$
```

```
    229 046454  162716  000004          15$:    SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    230 046460  000207                  16$:    RTS     PC              ;RETURN
    231
    232 046462  000000                  300$:   .WORD   0               ;ERROR FLAGS
    233
```

```
                        .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
                        ;       ???

                        CNTCLR:
046464  010046                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
046466  010146                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
046470  013746  000004        MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
046474  013746  000006        MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
046500  012737  046540  000004    MOV     #10$,ERRVEC     ;SETUP FOR BUS TIMEOUT
046506  012737  000300  000006    MOV     #PR6,ERRVEC+2
046514  013700  001276            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
046520  012760  000040  000010    MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
046526  013701  001464            MOV     TSTQUE,R1       ;GET DEVICE UNDER TEST
046532  111160  000010            MOVB    (R1),RMCS2(R0)  ;SELECT DEVICE
046536  000412                    BR      20$

046540  022626                10$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
046542  062766  000004  000010    ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
046550  112776  000007  000010    MOVB    #7,@10(SP)      ;WRITE THE ERROR NUMBER
046556  162766  000002  000010    SUB     #2,10(SP)       ;ADJUST SP TO RETURN TO ERROR
046564                      20$:
046564  012637  000006            MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
046570  012637  000004            MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
046574  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
046576  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
046600  000207                    RTS     PC
```

```
                                .SBTTL  STATUS CHECK SUBROUTINES
                                ;;*****************************************************************
                                .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,DLC
                                ;

                                ;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

046602                          CLPSTS:

                                ;CLEAR USER'S ERROR CALL
046602  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR
046606  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
046612  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR

                                ;REPORT ERROR IF RMCS1 NOT INITIALIZED
046616  013737  001334  001142  4$:     MOV     RMCS1I,$BDDAT   ;VERIFY RMCS1
046624  042737  100000  001142          BIC     #SC,$BDDAT      ;IGNORE SPECIAL CONDITION
046632  012737  004200  001140          MOV     #DVA!RDY,$GDDAT ;EXPECT DVA & RDY
046640  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED, RECEIVED
046646  001413                          BEQ     5$              ;BRANCH IF EQUAL
046650  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
046654  112776  000126  000000          MOVB    #126,@(SP)      ;WITE ERROR NUMBER IN CALL
046662  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046666  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
046670  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
046674  000240                          NOP

                                ;REPORT ERROR IF RMBA NOT RESET
046676  005037  001140          5$:     CLR     $GDDAT          ;VERIFY RMBA IS ZERO
046702  013737  001340  001142          MOV     RMBAI,$BDDAT
046710  001413                          BEQ     7$              ;BRANCH IF ZERO
046712  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
046716  112776  000127  000000          MOVB    #127,@(SP)      ;WITE ERROR NUMBER IN CALL
046724  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046730  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
046732  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
046736  000240                          NOP

                                ;REPORT ERROR IF RMCS2 NOT INITIALIZED
046740  013737  001344  001142  7$:     MOV     RMCS2I,$BDDAT   ;VERIFY RMCS2.
046746  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
046750  005046                          CLR     -(SP)           ;EXPECT IR & UNIT NUMBER
046752  013701  001464                  MOV     TSTQUE,R1       ;R1 = ADDRESS OF TEST QUE
046756  111116                          MOVB    (R1),(SP)
```

```
 58 046760  052716  000100                    BIS     #IR,(SP)
 59 046764  012637  001140                    MOV     (SP)+,$GDDAT
 60 046770  01260'                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 61 046772  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 62 047000  001413                            BEQ     9$              ;BRANCH IF EQUAL
 63 047002  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 64 047006  112776  000130  000000            MOVB    #130,@(SP)      ;WITE ERROR NUMBER IN CALL
 65 047014  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 66 047020  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 67 047022  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 68 047026  000240                            NOP
 69                                 ;REPORT ERROR IF RMER1 NOT RESET-IGNORE UNS
 70 047030  005037  001140          9$:       CLR     $GDDAT          ;VERIFY RMER1
 71 047034  013737  001350  001142            MOV     RMER11,$BDDAT
 72 047042  042737  040000  001142            BIC     #UNS,$BDDAT     ;IGNORE UNSAFE
 73 047050  001413                            BEQ     13$             ;BRANCH IF ZERO
 74 047052  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 75 047056  112776  000131  000000            MOVB    #131,@(SP)      ;WITE ERROR NUMBER IN CALL
 76 047064  162716  00000?                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 77 047070  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 78 047072  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 79 047076  000240                            NOP
 80                                 ;REPORT ERROR IF RMMR1 NOT INITIALIZED-IGNORE WC,LS,LST
 81 047100  013737  001360  001142  13$:       MOV     RMMR11,$BDDAT   ;VERIFY RMMR
 82 047106  042737  000046  001142            BIC     #WC.LS!LST,$BDDAT       ;IGNORE WORD CLOCK, SCT, TRK
 83 047114  012737  000010  001140            MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA BIT
 84 047122  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
 85 047130  001413                            BEQ     17$             ;BRANCH IF 0
 86 047132  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 87 047136  112776  000133  000000            MOVB    #133,@(SP)      ;WITE ERROR NUMBER IN CALL
 88 047144  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 89 047150  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 90 047152  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 91 047156  000240                            NOP
 92                                 ;REPORT AN ERROR IF RMEC2 IS NOT RESET
 93 047160  005037  001140          17$:      CLR     $GDDAT          ;EXPECT ZEROS
 94 047164  013737  001402  001142            MOV     RMEC21,$BDDAT   ;VERIFY RMEC2=0
 95 047172  001413                            BEQ     19$
 96 047174  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 97 047200  112776  000135  000000            MOVB    #135,@(SP)      ;WITE ERROR NUMBER IN CALL
 98 047206  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 99 047212  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
100 047214  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
101 047220  000240                            NOP
102                                ;REPORT ERROR IF RMMR2 NOT INITIALIZED-IGNORE RQA,RQB
103 047222  013737  001374  001142  19$:       MOV     RMMR21,$BDDAT   ;VERIFY RMMR2
104 047230  042737  140000  001142            BIC     #RQA!RQB,$BDDAT
105 047236  012737  011777  001140            MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
106 047244  023737  001140  001142            CMP     $GDDAT,$BDDAT
107 047252  001413                            BEQ     21$
108 047254  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
109 047260  112776  000136  000000            MOVB    #136,@(SP)      ;WITE ERROR NUMBER IN CALL
110 047266  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
111 047272  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
112 047274  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
113 047300  000240                            NOP
114                                ;REPORT ERROR IF RMER2 NOT RESET-IGNORE SKI,DVC
```

```
                                21$:    CLR     $GDDAT          ;EXPECT ALL ZEROS
                                        MOV     RMER21,$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$:   MOV     RMDS1,$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

145
```

```
                                .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
                                
047462                          ACKSTS:
                                
                                ;CLEAR USER'S ERROR CALL
047462  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
047466  105076  000000                  CLRB    @(SP)           ;CLEAR LOW ORDER BYTE
047472  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK
                                
                                ;REPORT AN ERROR  IF "VV" IS 0
047476  032737  000100  001346          BIT     #VV,RMDSI       ;IS VOLUME VALID SET??
047504  001024                          BNE     1$              ;YES.!
047506  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
047514  052737  000100  001140          BIS     #VV,$GDDAT
047522  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
047530  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
047534  112776  000155  000000          MOVB    #155,@(SP)      ;WRITE NUMBER IN ERROR CALL
047542  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
047546  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR
047550  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
047554  000240                          NOP
047556                          1$:
                                
                                ;REPORT AN ERROR IF "MOL" IS 0
047556  032737  010000  001346          BIT     #MOL,RMDSI      ;IS MOL SET??
047564  001024                          BNE     2$              ;YES!!
047566  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
047574  052737  010000  001140          BIS     #MOL,$GDDAT
047602  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
047610  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
047614  112776  000041  000000          MOVB    #41,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
047622  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
047626  004736                          JSR     PC,@(SP)+       ;REPORT TH ERROR
047630  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO BRANCH
047634  000240                          NOP
047636                          2$:
                                
                                ;SEE IF "UNS","OPI","RMR","ILR", OR "ILF" IS SET
047636  032737  060007  001350          BIT     #UNS!OPI!RMR!ILR!ILF,RMER1I
047644  001570                          BEQ     7$
                                
                                ;REPORT AN ERROR IF "UNS" IS SET
047646  032737  040000  001350          BIT     #UNS,RMER1I     ;WAS UNS SET??
047654  001424                          BEQ     3$              ;NO!!
047656  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

```
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #UNS,$GDDAT
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #42,@(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
                                        3$:

                                        ;REPORT ANY OPI ERROR
                                                BIT     #OPI,RMER1I     ;WAS OPI SET ??
                                                BEQ     4$              ;NO!!
                                                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #OPI,$GDDAT
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #43,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        4$:

                                        ;REPORT ANY RMR ERROR
                                                BIT     #RMR,RMER1I     ;WAS RMR SET??
                                                BEQ     5$              ;NO!!
                                                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #RMR,$GDDAT
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #44,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        5$:

                                        ;REPORT ANY     ILR ERROR
                                                BIT     #ILR,RMER1I     ;WAS ILR SET??
                                                BEQ     6$              ;NO!!
                                                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #ILR,$GDDAT
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #45,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        6$:

                                        ;REPORT ANY     ILF ERROR
                                                BIT     #ILF,RMER1I     ;WAS ILF SET??
                                                BEQ     7$              ;NO!!
                                                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
```

```
    115 050172  042737  000001  001140          BIC     #ILF,$GDDAT
    116 050200  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    117 050204  112776  000046  000000          MOVB    #46,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
    118 050212  162716  0000·2                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    119 050216  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
    120 050220  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
    121 050224  000240                          NOP
    122 050226                          7$:
    123
    124                                 ;AUGMENT RETURN ADDRESS IF ERROR WAS FOUND
    125 050226  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    126 050232  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
    127 050236  001403                          BEQ     8$              ;NO!!
    128 050240  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
    129 050244  000402                          BR      9$
    130 050246  162716  000004          8$:     SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
    131 050252  000240                  9$:     NOP
    132 050254  000207                          RTS     PC
    133
```

```
                                .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

050256                          RCLSTS:

                                ;CLEAR USER'S ERROR NUMBER
050256  062716  000004                  ADD     #4,(SP)
050262  105076  000000                  CLRB    @(SP)           ;CLEAR USER'S ERROR CALL
050266  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO BRANCH


                                ;SEE IF "PAR" OR "ILF" OR "OPI" OR "IAE" IS SET
050272  032737  022011  001350          BIT     #OPI!PAR!ILF!IAE,RMER1I
050300  001553                          BEQ     4$              ;NONE ARE SET - GO TO NEXT CHECK

                                ;REPORT ANY MASSBUS CONTROL BUS PARITY ERROR, I.E.,
                                ;"PAR" = 1 AND "DPE" = 0
050302  032737  000010  001350          BIT     #PAR,RMER1I     ;WAS "PAR" SET??
050310  001430                          BEQ     1$              ;NO!!
050312  032737  000010  001376          BIT     #DPE,RMER2I     ;WAS "DPE" SET??
050320  001024                          BNE     1$              ;YES - NOT A REGISTER ERROR
050322  013737  001350  001140          MOV RMER1I,$GDDAT       ;EXPECTED STATUS
050330  042737  000010  001140          BIC     #PAR,$GDDAT
050336  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
050344  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL.
050350  112776  000050  000000          MOVB    #50,@(SP)       ;WRITE ERROR NUMBER IN CALL
050356  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050362  004736                          JSR     PC,@(SP)+       ;GO REPORT ERROR
050364  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
050370  000240                          NOP
050372                          1$:

                                ;REPORT ANY "ILF" ERROR
050372  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS "ILF" SET??
050400  001424                          BEQ     2$              ;NO!!
050402  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
050410  042737  000001  001140          BIC     #ILF,$GDDAT
050416  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
050424  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
050430  112776  000071  000000          MOVB    #71,@(SP)       ;WRITE ERROR NUMBER IN CALL
050436  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050442  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
050444  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
050450  000240                          NOP
050452                          2$:

```

```
                                        ;REPORT ANY "OPI" ERROR AS
                                        ;       . OPI DUE TO "MOL" = 0
                                        ;       . OPI BECAUSE ON CYLINDER LATCH DIDN'T RESET
050452  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET??
050460  001433                          BEQ     31$             ;NO!!
050462  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
050470  042737  020000  001140          BIC     #OPI,$GDDAT
050476  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
050504  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
050510  112776  000072  000000          MOVB    #72,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
050516  032737  010000  001346          BIT     #MOL,RMDSI      ;WAS "MOL" = 0??
050524  001403                          BEQ     3$              ;YES!!
050526  112776  000073  000000          MOVB    #73,@(SP)       ;NO - CHANGE ERROR NUMBER
050534  162716  000002          3$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050540  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
050542  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
050546  000240                          NOP
050550                          31$:

                                        ;REPORT AN ERROR IF "IAE" IS SET
050550  032737  002000  001350          BIT     #IAE,RMER1I     ;IS "IAE" SET??
050556  001424                          BEQ     4$              ;NO!!
050560  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
050566  042737  002000  001140          BIC     #IAE,$GDDAT
050574  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
050602  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
050606  112776  000070  000000          MOVB    #70,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
050614  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050620  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
050622  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
050626  000240                          NOP
050630                          4$:

                                        ;SEE IF "SKI" OR "IVC" OR "DVC" IS SET
050630  032737  050200  001376          BIT     #SKI!IVC!DVC,RMER2I
050636  001517                          BEQ     8$              ;NONE OF THE BITS ARE SET


                                        ;REPORT ANY "IVC" ERROR AS
                                        ;       .IVC WITH VV = 0
                                        ;       .ERRONEOUS IVC ERROR
050640  032737  010000  001376          BIT     #IVC,RMER2I     ;WAS IVC SET??
050646  001433                          BEQ     6$              ;NO!!
050650  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
050656  042737  010000  001140          BIC     #IVC,$GDDAT
050664  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
050672  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
050676  112776  000074  000000          MOVB    #74,@(SP)       ;WRITE ERROR NUMBER IN CALL
050704  032737  000100  001346          BIT     #VV,RMDSI       ;WAS VV = 0??
050712  001403                          BEQ     5$              ;YES!!
050714  112776  000075  000000          MOVB    #75,@(SP)       ;NO - CHANGE ERROR NUMBER
050722  162716  000002          5$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050726  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
050730  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
050734  000240                          NOP
050736                          6$:

```

```
                                        ;REPORT ANY "SKI" ERROR
050736  032737  040000  001376                BIT     #SKI,RMER2I     ;WAS SKI SET??
050744  001424                              BEQ     7$              ;NO!!
050746  013737  001376  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
050754  042737  040000  001140              BIC     #SKI,$GDDAT
050762  013737  001376  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
050770  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
050774  112776  000076  000000              MOVB    #76,@(SP)       ;WRITE ERROR NUMBER
051002  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051006  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
051010  162716  000010                      SUB     #10.(SP)        ;MOVE SP TO BRANCH
051014  000240                              NOP
051016                                  7$:

                                        ;REPORT ANY "DVC" ERROR
051016  032737  000200  001376                BIT     #DVC,RMER2I     ;WAS "DVC" SET??
051024  001424                              BEQ     8$              ;NO!!
051026  013737  001376  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
051034  042737  000200  001140              BIC     #DVC,$GDDAT
051042  013737  001376  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
051050  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
051054  112776  000077  000000              MOVB    #77,@(SP)       ;WRITE ERROR NUMBER
051062  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051066  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
051070  162716  000010                      SUB     #10.(SP)        ;MOVE SP TO USER'S BRANCH
051074  000240                              NOP
051076                                  8$:

                                        ;SEE IF "PIP" AND "OM" ARE 0, AND "ATA","MOL" AND "VV" ARE 1
051076  013746  001346                      MOV     RMDSI,-(SP)     ;PUT RMDS ON STACK
051102  042716  047676                      BIC     #^C<PIP!MOL!VV OM!ATA>,(SP)
051106  022726  110100                      CMP     #ATA!MOL!VV,(SP)+
051112  001002                              BNE     85$
051114  000137  051530                      JMP     13$
051120                                  85$:

                                        ;REPORT AN ERROR IF MOL = 0 AND OPI = 0, I.E., MEDIUM WENT OFF
                                        ;LINE AFTER RECALIBRATE WAS INITIATED
051120  032737  010000  001346                BIT     #MOL,RMDSI      ;DID MOL DROP??
051126  001030                              BNE     9$              ;NO!!
051130  032737  020000  001350              BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
051136  001024                              BNE     9$              ;YES - DON'T REPORT MOL=0
051140  013737  001346  001140              MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
051146  052737  010000  001140              BIS     #MOL,$GDDAT
051154  013737  001346  001142              MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
051162  062716  000004                      ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
051166  112776  000100  000000              MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
051174  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051200  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
051202  162716  000010                      SUB     #10.(SP)        ;MOVE SP BACK TO USER'S BRANCH
051206  000240                              NOP
051210                                  9$:

                                        ;REPORT AN ERROR IF "VV" = 0 AND "IVC" = 0
051210  032737  000100  001346                BIT     #VV,RMDSI       ;DID "VV" DROP??
051216  001030                              BNE     10$             ;NO!!
051220  032737  010000  001376              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
```

```
229 051530                          13$:
230
231                                 ;SEE IF "ILR" OR "RMR" OR "UNS" IS SET
232 051530  032737  040006  001350          BIT     #ILR.RMR'UNS,RMER1I
233 051536  001514                          BEQ     16$
234
235                                 ;REPORT AN ERROR IF "ILR" IS SET
236 051540  032737  000002  001350          BIT     #ILR,RMER1I     ;WAS ILR SET DURING RECALIBRATE??
237 051546  001424                          BEQ     14$             ;NO!!
238 051550  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
239 051556  042737  000002  001140          BIC     #ILR,$GDDAT
240 051564  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
241 051572  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
242 051576  112776  000105  000000          MOVB    #105,@(SP)      ;WRITE ERROR NUMBER IN CALL
243 051604  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
244 051610  004736                          JSR     PC,@(SP)+
245 051612  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
246 051616  000240                          NOP
247 051620                          14$:
248
249                                 ;REPORT AN ERROR IF "RMR" IS SET
250 051620  032737  000004  001350          BIT     #RMR,RMER1I     ;WAS RMR SET??
251 051626  001424                          BEQ     15$             ;NO.!
252 051630  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
253 051636  042737  000004  001140          BIC     #RMR,$GDDAT
254 051644  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
255 051652  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
256 051656  112776  000106  000000          MOVB    #106,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
257 051664  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
258 051670  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
259 051672  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
260 051676  000240                          NOP
261 051700                          15$:
262
263                                 ;REPORT AN ERROR IF "UNS" IS SET AND "DVC" IS 0
264 051700  032737  040000  001350          BIT     #UNS,RMER1I     ;WAS UNSAFE ON??
265 051706  001430                          BEQ     16$             ;NO!!
266 051710  032737  000200  001376          BIT     #DVC,RMER2I     ;WAS THERE A DEVICE CHECK??
267 051716  001024                          BNE     16$             ;YES - DON'T REPORT UNSAFE
268 051720  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
269 051726  042737  040000  001140          BIC     #UNS,$GDDAT
270 051734  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
271 051742  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
272 051746  112776  000107  000000          MOVB    #107,@(SP)      ;WRITE ERROR NUMBER
273 051754  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
274 051760  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
275 051762  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
276 051766  000240                          NOP
277 051770                          16$:
278
279                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
280 051770  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
281 051774  105776  000000                  TSTB    @(SP)           ;WAS AN ERROR REPORTED??
282 052000  001403                          BEQ     17$             ;NO!!
283 052002  062716  000004                  ADD     #4,(SP)         ;YES - AUGMENT SP RETURN
284 052006  000402                          BR      18$
285 052010  162716  000004          17$:    SUB     #4,(SP) ;NO ERROR - RETURN SP TO BRANCH
```

```
286 052014  000240                18$:    NOP
287 052016  000207                        RTS     PC              ;STATUS CECK IS COMPLETE
288
```

```
                                        .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
                                        ;
052020                                  DRVSTS:
                                        ;
                                        ;CLEAR USER'S ERROR CALL
052020  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
052024  105076  000000                          CLRB    @(SP)           ;CLEAR ERROR CALL
052030  162716  000004                          SUB     #4,(SP)         ;MOVE SP TO USER'S BRANCH
                                        ;REPORT ERROR IF RMCS1 NOT INITIALIZED
052034  013737  001534  001142          4$:     MOV     RMCS1I,$BDDAT   ;CHECK RMCS1
052042  042737  173700  001142                  BIC     #^C<DVA!FNCMSK>,$BDDAT  ;CLEAR DONT CARES
052050  012737  004010  001140                  MOV     #DVA!DRVCLR,$GDDAT      ;EXPECT DVA
052056  023737  001140  001142                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
052064  001443                                  BEQ     6$              ;BRANCH IF EQUAL
052066  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
052072  112776  000141  000000                  MOVB    #141,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
052100  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052104  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
052106  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
052112  000240                                  NOP
                                        ;REPORT ERROR IF RMDS NOT INITIALIZED
052114  013737  001346  001142          5$:     MOV     RMDSI,$BDDAT    ;CHECK RMDS
052122  042737  021101  001142                  BIC     #PGM!OM!VV!PIP,$BDDAT   ;CLEAR DONT CARES
052130  012737  010600  001140                  MOV     #MOL!DPR!DRY,$GDDAT     ;EXPECT DRY & DPR
052136  023737  001140  001142                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
052144  001413                                  BEQ     6$              ;BRANCH IF EQUAL
052146  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
052152  112776  000142  000000                  MOVB    #142,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
052160  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052164  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
052166  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
052172  000240                                  NOP
                                        ;REPORT ERROR IF RMER1 NOT INITIALIZED
052174  005037  001140                  6$:     CLR     $GDDAT          ;EXPECT 0'S
052200  013737  001350  001142                  MOV     RMER1I,$BDDAT   ;CHECK RMER1
052206  001413                                  BEQ     8$              ;BRANCH IF EQUAL
052210  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
052214  112776  000143  000000                  MOVB    #143,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
052222  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052226  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
052230  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
052234  000240                                  NOP
                                        ;REPORT ERROR IF ATA NOT INITIALIZED
052236  013737  001352  001142          8$:     MOV     RMAS1,$BDDAT    ;CHECK ATTENTION BIT
052244  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052246  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
052250  013701  001464                          MOV     TSTQUE,R1
052254  116102  000001                          MOVB    1(R1),R2
052260  042702  177400                          BIC     #^CATNMSK,R2
052264  005102                                  COM     R2
052266  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$:
```

```
115
116 052572  062716  000004                ;AUGMENT RETURN ADDRESS IF ANY ERROR WAS FOUND
117 052576  105776  000000                        ADD     #4,(SP)          ;MOVE SP TO ERROR CALL
118 052602  001403                                TSTB    @(SP)            ;WAS AN ERROR DETECTED??
119 052604  062716  000004                        BEQ     21$              ;NO!!
120 052610  000402                                ADD     #4,(SP)          ;YES - MOVE SP TO ERROR RETURN
121 052612  162716  000004                        BR      23$
122 052616  000240                  21$:  23$:    SUB     #4,(SP) ;MOVE SP BACK TO NO ERROR RETURN
123 052620  000207                                NOP
124                                               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 REFORT 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

052622                          DTASTS:

                                ;CLEAR USER'S ERROR CALL AND ERROR FLAGS
052622  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
052626  105076  000000                CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF TRAP
052632  162716  000004                SUB     #4,(SP)         ;RESTORE SP TO NO ERROR
052636  005037  056216                CLR     500$            ;CLEAR ERROR FLAGS

                                ;REPORT ANY CONTROL BUS PARITY ERROR WHILE READING REMOTE REGISTERS,
                                ;I.E., MCPE = 1
052642  032737  020000  001334        BIT     #MCPE,RMCS1I    ;WAS THERE A PARITY ERROR??
052650  001422                      BEQ     10$             ;NO!!
052652  013737  001334  001140      MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
052660  042737  020000  001140      BIC     #MCPE,$GDDAT
052666  013737  001334  001142      MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
052674  062716  000004              ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052700  112776  000013  000000      MOVB    #13,@(SP)       ;WRITE ERROR NUMBER
052706  162716  000002              SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
052712  004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
052714  000466                      BR      30$
052716                          10$:

                                ;REPORT ANY CONTROL BUS PARITY ERROR WHILE WRITTING REMOTE REGISTERS,
                                ;I.E., PAR = 1 AND DPE = 0
052716  032737  000010  001350        BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
052724  001435                      BEQ     20$             ;NO!!
052726  032737  000010  001376        BIT     #DPE,RMER2I     ;DATA PARITY ERROR ?
052734  001031                      BNE     20$             ;YES!!
052736  013737  001350  001140      MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
052744  042737  000010  001140      BIC     #PAR,$GDDAT
052752  013737  001350  001142      MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
052760  062716  000004              ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
052764  112776  000050  000000      MOVB    #50,@(SP)       ;WRITE ERROR NUMBER
052772  032737  001000  001344      BIT     #MXF,RMCS2I     ;DID MXF GET SET??
053000  001003                      BNE     15$             ;YES!!
053002  112776  000274  000000      MOVB    #274,@(SP)      ;NO - CHANGE ERROR NUMBER
053010  162716  000002          15$:SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053014  004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
053016  000425                      BR      30$

                                ;LOOK FOR ANY ERRORS WHICH MAY HAVE OCCURRED DURING COMMAND INITIATION OR
```

```
                                        ;MECHANICAL POSITIONING
                                        ;
                                        ;FIRST TEST MXF WHICH WOULD INDICATE COMPOSITE ERROR SET WHEN FUNCTION
                                        ;CODE AND GO BIT WERE LOADED
053020                                  20$:
053020  032737  001000  001344                BIT     #MXF,RMCS2I     ;WAS "MISSED TRANSFER" SET??
053026  001425                            BEQ     40$             ;NO.!
053030  013737  001344  001140            MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
053036  042737  001000  001140            BIC     #MXF,$GDDAT
053044  013737  001344  001142            MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
053052  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053056  112776  000275  000000            MOVB    #275,@(SP)      ;WRITE ERROR NUMBER
053064  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053070  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
053072                                  30$:
                                        ;
                                        ;RESTORE SP TO NO ERROR RETURN AND BYPASS FURHTER STATUS CHECKING
053072  162716  000010                    SUB     #10,(SP)        ;MOVE SP TO NO ERROR
053076  000137  056170                    JMP     380$            ;SKIP TO END OF SUB
                                        ;
053102                                  40$:
                                        ;
                                        ;REPORT AN ERROR IF "OPI" ERROR OCCURRED DUE TO "MOL" = 0, OR IF "OPI"
                                        ;AND "MOL" ARE SET, BUT "VV" IS RESET, INDICATING AN INTERMITTENT
                                        ;"MOL"
053102  032737  020000  001350            BIT     #OPI,RMER1I     ;IS "OPI" SET??
053110  001447                            BEQ     60$             ;NO!!
053112  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053120  042737  020000  001140            BIC     #OPI,$GDDAT
053126  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053134  032737  010000  001346            BIT     #MOL,RMDSI      ;WAS MEDIUM OFF LINE??
053142  001404                            BEQ     45$             ;YES!!
053144  032737  000100  001346            BIT     #VV,RMDSI       ;WAS "MOL" INTERMITTENT??
053152  001013                            BNE     50$             ;NO!!
053154  062716  000004            45$:    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053160  112776  000276  000000            MOVB    #276,@(SP)      ;WRITE ERROR NUMBER IN CALL
053166  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053172  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
053174  162716  000010                    SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
053200  000413                            BR      60$
053202                                  50$:
                                        ;
                                        ;REPORT "OPI" ERROR, WHICH IS DUE TO "ON CYLINDER" NOT DROPPING OR
                                        ;"RUN" TIMEOUT (20 MS) OR SEARCH TIMEOUT (50 MS)
053202  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053206  112776  000277  000000            MOVB    #277,@(SP)      ;WRITE ERROR NUMBER IN CALL
053214  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053220  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
053222  162716  000010                    SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
053226  000240                            NOP
053230                                  60$:
                                        ;
                                        ;LOOK FOR "IVC" ERROR DURING COMMAND INITIATION
053230  032737  010000  001376            BIT     #IVC,RMER2I     ;WAS THERE AN "IVC" ERROR??
053236  001432                            BEQ     70$             ;NO!!
                                        ;REPORT "IVC" ERROR DUE TO "VV" = 0, OR REPORT ERRONEOUS "IVC" ERROR
053240  013737  001376  001140            MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
```

```
053246  042737  010000  001140          BIC     #IVC,$GDDAT
053254  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
053262  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
053266  112776  000300  000000          MOVB    #300,@(SP)      ;WRITE ERROR NUMBER IN CALL
053274  032737  000100  001346          BIT     #VV,RMDSI       ;WAS VOLUME VALID??
053302  001403                          BEQ     65$             ;NO!!
053304  112776  000301  000000          MOVB    #301,@(SP)      ;CHANGE ERROR NUMBER
053312  162716  000002          65$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053316  004736                          JSR     PC,@(SP)+       ;REPORT "IVC" ERROR AND RETURN
053320  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
053324                          70$:

                        ;SEE IF "ILF" OR "RMR" IS SET
053324  032737  000007  001350          BIT     #ILR!ILF!RMR, RMER1I
053332  001510                          BEQ     100$            ;NO ERRORS DETECTED
                        ;REPORT AN ERROR IF "ILR" IS SET
053334  032737  000002  001350          BIT     #ILR,RMER1I     ;WAS "ILR" DETECTED??
053342  001424                          BEQ     80$             ;NO!!
053344  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053352  042737  000002  001140          BIC     #ILR,$GDDAT
053360  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053366  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053372  112776  000302  000000          MOVB    #302,@(SP)      ;WRITE ERROR NUMBER IN CALL
053400  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053404  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
053406  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
053412  000240                          NOP
053414                          80$:

                        ;REPORT AN ERROR IF "ILF" IS SET
053414  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS "ILF" DETECTED??
053422  001424                          BEQ     90$             ;NO!!
053424  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053432  042737  000001  001140          BIC     #ILF,$GDDAT
053440  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053446  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053452  112776  000303  000000          MOVB    #303,@(SP)      ;WRITE ERROR NUMBER IN CALL
053460  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053464  004736                          JSR     PC,@(SP)+       ;REPORT ERORR AND RETURN
053466  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
053472  000240                          NOP
053474                          90$:

                        ;REPORT AN ERROR IF "RMR" IS SET
053474  032737  000004  001350          BIT     #RMR,RMER1I     ;WAS "RMR" DETECTED??
053502  001424                          BEQ     100$            ;NO!!
053504  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053512  042737  000004  001140          BIC     #RMR,$GDDAT
053520  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053526  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053532  112776  000304  000000          MOVB    #304,@(SP)      ;WRITE ERROR NUMBER IN CALL
053540  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
053544  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
053546  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
053552  000240                          NOP
053554                          100$:
                        ;DETERMINE WHETHER OR NOT "IAE" SHOULD BE SET AND CHECK FOR ERROR
```

```
053554  012737  002000  001140          MOV     #IAE,$GDDAT       ;SETUP FOR "IAE" = 1
053562  052737  040000  056216          BIS     #SKI,500$         ;SETUP FOR "SKI" = 1
053570  023727  001444  001466          CMP     RMDCO,#822.       ;GREATER THAN LAST CYLINDER ?
053576  101025                          BHI     110$              ;YES - CYLINDER IS INVALID
053600  042737  040000  056216          BIC     #SKI,500$         ;RESET SKI FLAG

053606  123737  001417  001333          CMPB    RMDAO+1,LSTRK+1   ;GREATER THAN LAST TRACK ?
053614  101016                          BHI     110$              ;YES - TRACK IS INVALID

053616  123727  001416  000035          CMPB    RMDAO,#29.        ;IS SECTOR > 29. ?
053624  101410                          BLOS    105$              ;NO
053626  032737  010000  001442          BIT     #FMT16,RMOFO      ;18 BIT FORMAT ?
053634  001406                          BEQ     110$              ;YES - SECTOR IS INVALID FOR 18 BIT MODE
053636  123727  001416  000037          CMPB    RMDAO,#31.        ;IS SECTOR > 31. ?
053644  101002                          BHI     110$              ;YES - SECTOR IS INVALID
053646  005037  001140          105$:   CLR     $GDDAT            ;"IAE" SHOULD = 0

053652  013737  001350  001142  110$:   MOV     RMER1I,$BDDAT     ;GET RECEIVED STATUS
053660  042737  175777  001142          BIC     #^CIAE,$BDDAT
053666  023737  001140  001142          CMP     $GDDAT,$BDDAT     ;IS "IAE" STATUS OK??
053674  001004                          BNE     115$              ;NO!.
053676  042737  040000  056216          BIC     #SKI,500$         ;IAE OK - SKI SHOULD BE 0
053704  000412                          BR      120$
053706  062716  000004          115$:   ADD     #4,(SP)           ;MOVE SP TO USER'S ERROR CALL
053712  112776  000305  000000          MOVB    #305,@(SP)        ;WRITE ERROR NUMBER
053720  162716  000002                  SUB     #2,(SP)           ;MOVE SP TO RETURN IF ERROR
053724  004736                          JSR     PC,@(SP)+         ;REPORT ERROR AND RETURN
053726  162716  000010                  SUB     #10,(SP)          ;MOVE SP TO NO ERROR
053732                          120$:


                                ;REPORT AN ERROR IF "SKI" IS SET AND "IAE" STATUS WAS OK
053732  013737  001376  001142          MOV     RMER2I,$BDDAT     ;RECEIVED STATUS
053740  042737  137777  001142          BIC     #^CSKI,$BDDAT
053746  013737  056216  001140          MOV     500$,$GDDAT       ;EXPECTED STATUS
053754  042737  137777  001140          BIC     #^CSKI,$GDDAT
053762  032737  040000  001376          BIT     #SKI,RMER2I       ;WAS "SKI" SET??
053770  001417                          BEQ     140$              ;NO!!
053772  032737  040000  056216          BIT     #SKI,500$         ;WAS SKI CAUSED BY IAE = 0??
054000  001032                          BNE     150$              ;YES - DON'T REPORT SKI
054002  062716  000004                  ADD     #4,(SP)           ;MOVE SP TO USERS ERROR CALL
054006  112776  000306  000000          MOVB    #306,@(SP)        ;WRITE ERROR NUMBER
054014  162716  000002                  SUB     #2,(SP)           ;MOVE SP TO RETURN IF ERROR
054020  004736                          JSR     PC,@(SP)+         ;REPORT ERROR AND RETURN
054022  162716  000010                  SUB     #10,(SP)          ;MOVE SP TO NO ERROR
054026  000417                          BR      150$

054030                          140$:

                                ;REPORT AN ERROR IF SKI = 0 AND IAE WAS NOT DETECTED
054030  032737  040000  056216          BIT     #SKI,500$         ;SHOULD SKI BE SET??
054036  001413                          BEQ     150$              ;NO!!
054040  062716  000004                  ADD     #4,(SP)           ;MOVE SP TO USER'S ERROR CALL
054044  112776  000307  000000          MOVB    #307,@(SP)        ;WRITE ERROR NUMBER IN CALL
054052  162716  000002                  SUB     #2,(SP)           ;MOVE SP TO RETURN IF ERROR
054056  004736                          JSR     PC,@(SP)+         ;REPORT ERROR AND RETURN
054060  162716  000010                  SUB     #10,(SP)          ;RESTORE SP TO NO ERROR
054064  000240                          NOP
```

```
054066                          150$:

                                ;LOOK FOR "LSC" OR "LBC" OR "DVC" IN ERROR REGISTER #2
054066  032737  006200  001376          BIT     #LSC!LBC!DVC,RMER2I
054074  001512                        BEQ     180$            ;NO ERRORS SET

                                ;REPORT ANY DEVICE FAULT, I.E., "DVC" = 1
054076  032737  000200  001376          BIT     #DVC,RMER2I     ;IS "DVC" = 1??
054104  001424                        BEQ     160$            ;NO!!
054106  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
054114  042737  000200  001140          BIC     #DVC,$GDDAT
054122  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
054130  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
054134  112776  000310  000000          MOVB    #310,@(SP)      ;WRITE ERROR NUMBER IN CALL
054142  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054146  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054150  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
054154  000240                          NOP
054156                          160$:

                                ;REPORT LOSS OF BIT CLOCK, I.E.; "LBC" = 1, IF "MOL" = 1
054156  032737  002000  001376          BIT     #LBC,RMER2I     ;IS LBC SET??
054164  001430                        BEQ     170$            ;NO!!
054166  032737  010000  001346          BIT     #MOL,RMDS1      ;WAS LBC ERROR BY MOL = 0
054174  001424                        BEQ     170$            ;YES!!
054176  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
054204  042737  002000  001140          BIC     #LBC,$GDDAT
054212  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
054220  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054224  112776  000311  000000          MOVB    #311,@(SP)      ;WRITE ERROR NUMBER IN CALL
054232  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054236  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054240  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
054244  000240                          NOP
054246                          170$:

                                ;REPORT LOS OF SYSTEM CLOCK, I.E., "LSC" = 1
054246  032737  004000  001376          BIT     #LSC,RMER2I     ;IS "LSC" = 1??
054254  001422                        BEQ     180$            ;NO!!
054256  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
054264  042737  004000  001140          BIC     #LSC,$GDDAT
054272  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
054300  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054304  112776  000312  000000          MOVB    #312,@(SP)      ;WRITE ERROR NUMBER
054312  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054314  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
054320  000240                          NOP
054322                          180$:

                                ;LOOK FOR "UNS" OR "DTE" OR "WLE" IN ERROR REGISTER #1
054324  032737  054000  001350          BIT     #UNS!DTE!WLE,RMER1I
054332  001527                        BEQ     220$            ;NO BITS SET
                                ;REPORT "UNS" ERROR IF "DVC" = 0
054334  032737  040000  001350          BIT     #UNS,RMER1I     ;IS "UNS" SET??
054342  001427                        BEQ     190$            ;NO!!
054344  032737  000200  001376          BIT     #DVC,RMER2I     ;WAS "UNS" CAUSED BY "DVC"??
054352  001023                        BNE     190$            ;YES!!
```

```
054352  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054360  042737  040000  001140          BIC     #UNS,$GDDAT
054366  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054374  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
054400  112776  000313  000000          MOVB    #313,@(SP)      ;WRITE ERROR NUMBER
054406  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054412  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054414  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
054420                          190$:

                                ;REPORT ANY DRIVE TIMING ERROR, I.E., "DTE" = 1
054420  032737  010000  001350          BIT     #DTE,RMER1I     ;IS DTE SET??
054426  001423                          BEQ     200$            ;NO!!
054430  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054436  042737  010000  001140          BIC     #DTE,$GDDAT
054444  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054452  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054456  112776  000314  000000          MOVB    #314,@(SP)      ;WRITE ERROR NUMBER IN CALL
054464  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054470  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054472  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
054476                          200$:

                                ;REPORT AN ERROR IF WRITE LOCK ERROR IS SET.  SEE IF DRIVE IS NOT
                                ;WRITE PROTECTED, OR IF FUNCTION WAS NOT A WRITE
054476  032737  004000  001350          BIT     #WLE,RMER1I     ;WAS "WLE" SET??
054504  001441                          BEQ     220$            ;NO!!
054506  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054514  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054522  052737  004000  001140          BIS     #WLE,$GDDAT
054530  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
054534  112776  000315  000000          MOVB    #315,@(SP)      ;WRITE ERROR NUMBER IN CALL
054542  032737  004000  001346          BIT     #WRL,RMDSI      ;WAS DRIVE WRITE PROTECTED??
054550  001404                          BEQ     205$            ;NO!!
054552  032737  000010  001410          BIT     #BIT3,RMCS10    ;WAS COMMAND A WRITE??
054560  001406                          BEQ     210$            ;YES!!
054562  112776  000316  000000  205$:   MOVB    #316,@(SP)      ;CHANGE ERROR NUMBER
054570  042737  004000  001140          BIC     #WLE,$GDDAT
054576  162716  000002          210$:   SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054602  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054604  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR

054610                          220$:

                                ;OMIT DATA ERROR CHECKS IF ANY PREVIOUS ERRORS HAVE BEEN DETECTED
054610  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
054614  105776  000000                  TSTB    @(SP)           ;WAS ERROR DETECTED??
054620  001404                          BEQ     225$            ;NO - DO DATA CHECKS
054622  162716  000004                  SUB     #4,(SP)         ;RESTORE SP
054626  000137  055630                  JMP     340$            ;SKIP DATA CHECKS
054632  162716  000004          225$:   SUB     #4,(SP)         ;RESTORE SP

                                ;CHECK HEADER ERRORS IF FUNCTION WAS NOT WRITE HEADER AND DATA, AND
                                ;IF HEADER COMPARE IS NOT INHIBITED
054636  013737  001410  056220          MOV     RMCS10,510$     ;STRIP AND STORE FUNCTION CODE
054644  042737  177700  056220          BIC     #^CFNCMSK,510$
054652  022737  000063  056220          CMP     #WH'GO,510$     ;WAS FUNCTION WRITE HEADER & DATA??
```

```
                                        BEQ     250$            ;YES - SKIP HEADER CHECKS
                                        BIT     #HCI,RMOFI      ;WAS HCI SET??
                                        BNE     250$            ;YES - SKIP HEADER CHECKS

                                ;SEE IF ANY HEADER ERRORS ARE SET, I.E., "FER" OR "HCRC" OR "HCE"
                                        BIT     #HCRC!FER!HCE,RMER1I
                                        BEQ     270$            ;NO ERRORS SET

                                ;REPORT HEADER CRC ERROR IF SET
                                        BIT     #HCRC,RMER1I    ;WAS HCRC SET??
                                        BEQ     230$            ;NO!!
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #HCRC,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
                                        MOVB    #317,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        BR      260$
                                230$:

                                ;REPORT FORMAT ERROR IF SET
                                        BIT     #FER,RMER1I     ;WAS "FER" SET??
                                        BEQ     240$            ;NO.!
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #FER,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
                                        MOVB    #320,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        BR      260$
                                240$:

                                ;REPORT HEADER COMPARE ERROR IF SET
                                        BIT     #HCE,RMER1I     ;WAS "HCE" SET??
                                        BEQ     270$            ;NO.!
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #HCE,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                        MOVB    #321,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        BR      260$
                                ;THERE SHOULD BE NO HEADER ERRORS BECAUSE
                                ;       .COMMAND WAS WRITE HEADER AND DATA, OR
                                ;       .HEADER COMPARE INHIBIT WAS SET
                                250$:   BIT     #HCE!FER!HCRC,RMER1I
                                        BEQ     270$            ;NO ERRORS WERE SET
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #HCE!FER!HCRC,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #322,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
```

```
                                ;IF COMMAND WAS A WRITE COMMAND, GO DO WRITE ERROR CHECKS, OTHERWISE
                                ;DO READ ERROR CHECKS
055170  032737  000010  056220          BIT     #BIT3,510$      ;WAS THIS A WRITE COMMAND?
055176  001002                          BNE     275$            ;NO!!
055200  000137  055416                  JMP     310$            ;GO DO WRITE STATUS CHECK
055204                          275$:

                                ;REPORT DATA CHECK IF SET
055204  032737  100000  001350          BIT     #DCK,RMER1I     ;DATA CHECK ERROR??
055212  001450                          BEQ     290$            ;NO!!
055214  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055222  042737  100000  001140          BIC     #DCK,$GDDAT
055230  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055236  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S EPROR
055242  112776  000323  000000          MOVB    #323,@(SP)      ;WRITE ERROR NUMBER
055250  032737  004000  001366          BIT     #ECI,RMOFI      ;WAS ECC CORRECTION DISABLED??
055256  001021                          BNE     280$            ;YES!!
055260  112776  000324  000000          MOVB    #324,@(SP)      ;CHANGE TO RECOVERABLE ERROR
055266  032737  000100  001350          BIT     #ECH,RMER1I     ;IS ERROR RECOVERABLE??
055274  001007                          BNE     276$            ;NO !!
                                ;DO NOT REPORT RECOVERABLE ERROR IF READ COMMAND
055276  032737  000020  056220          BIT     #BIT4,510$      ;WAS THIS A READ COMMAND ??
055304  001406                          BEQ     280$            ;NO !!
055306  162716  000004                  SUB     #4,(SP)         ;RESTORE SP
055312  000410                          BR      290$            ;SKIP ERROR - DATA WILL BE CORRECTED
055314  112776  000325  000000  276$:   MOVB    #325,@(SP)      ;CHANGE TO NON RECOVERABLE
055322  162716  000002          280$:   SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
055326  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
055330  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR

055334                          290$:

                                ;REPORT DATA BUS PARITY ERROR IF SET, I.E., MDPE = 1
055334  032737  000400  001344          BIT     #MDPE,RMCS2I    ;PARITY ERROR SET??
055342  001423                          BEQ     300$            ;NO!!
055344  013737  001344  001140          MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
055352  042737  000400  001140          BIC     #MDPE,$GDDAT
055360  013737  001344  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
055366  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
055372  112776  000326  000000          MOVB    #326,@(SP)      ;WRITE ERROR NUMBER
055400  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
055404  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
055406  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
055412  000137  055630          300$:   JMP     340$            ;SKIP WRITE STATUS CHECK

055416                          310$:

                                ;TEST TO SEE THAT OFFSET MODE WAS RESET; REPORT ERROR IF ''OM''   1
055416  032737  000001  001346          BIT     #OM,RMDS1       ;IS OFFSET ON??
055424  001423                          BEQ     320$            ;NO
055426  013737  001346  001140          MOV     RMDS1,$GDDAT    ;EXPECTED STATUS
055434  042737  000001  001140          BIC     #OM,$GDDAT
```

```
055442  013737  001346  001142          MOV     RMDSI,$BDDAT     ;RECEIVED STATUS
055450  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USER'S ERROR CALL
055454  112776  000327  000000          MOVB    #327,@(SP)       ;WRITE ERROR NUMBER IN CALL
055462  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO RETURN IF ERROR
055466  004736                          JSR     PC,@(SP)+        ;REPORT ERROR AND RETURN
055470  162716  000010                  SUB     #10,(SP)         ;MOVE SP TO NO ERROR
055474                          320$:

                                ;TEST FOR DATA BUS PARITY ERROR; REPORT ERROR IF "DPE" = 1
055474  032737  000010  001376          BIT     #DPE,RMER2I      ;DATA PARITY ERROR??
055502  001423                          BEQ     330$             ;NO!!
055504  013737  001376  001140          MOV     RMER2I,$GDDAT    ;EXPECTED STATUS
055512  042737  000010  001140          BIC     #DPE,$GDDAT
055520  013737  001376  001142          MOV     RMER2I,$BDDAT    ;RECEIVED STATUS
055526  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USER'S ERROR CALL
055532  112776  000330  000000          MOVB    #330,@(SP)       ;WRITE ERROR NUMBER
055540  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO RETURN IF ERROR
055544  004736                          JSR     PC,@(SP)+        ;REPORT ERROR AND RETURN
055546  162716  000010                  SUB     #10,(SP)         ;MOVE SP TO NO ERROR
055552                          330$:

                                ;TEST FOR WRITE CLOCK FAILURE; REPORT ERROR IF "WCF" = 1
055552  032737  000040  001350          BIT     #WCF,RMER1I      ;IS "WCF" SET??
055560  001423                          BEQ     340$             ;NO!!
055562  013737  001350  001140          MOV     RMER1I,$GDDAT    ;EXPECTED STATUS
055570  042737  000040  001140          BIC     #WCF,$GDDAT
055576  013737  001350  001142          MOV     RMER1I,$BDDAT    ;RECEIVED STATUS
055604  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USERS ERROR CALL
055610  112776  000331  000000          MOVB    #331,@(SP)       ;WRITE ERROR NUMBER
055616  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO RETURN IF ERROR
055622  004736                          JSR     PC,@(SP)+        ;REPORT ERROR AND RETURN
055624  162716  000010                  SUB     #10,(SP)         ;MOVE SP TO NO ERROR
055630                          340$:

                                ;REPORT "DATA LATE" ERROR IF "DLT" = 1
055630  032737  100000  001344          BIT     #DLT,RMCS2I      ;IS "DLT" SET??
055636  001423                          BEQ     350$             ;NO!!
055640  013737  001344  001140          MOV     RMCS2I,$GDDAT    ;EXPECTED STATUS
055646  042737  100000  001140          BIC     #DLT,$GDDAT
055654  013737  001344  001142          MOV     RMCS2I,$BDDAT    ;RECEIVED STATUS
055662  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USERS ERROR CALL
055666  112776  000332  000000          MOVB    #332,@(SP)       ;WRITE ERROR NUMBER
055674  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO RETURN IF ERROR
055700  004736                          JSR     PC,@(SP)+        ;REPORT ERROR AND RETURN
055702  162716  000010                  SUB     #10,(SP)         ;MOVE SP TO NO ERROR
055706                          350$:
                                ;LOOK FOR UNEXPECTED CHANGES IN DRIVE STATUS
055706  013746  001346                  MOV     RMDSI,-(SP)      ;STACK DRIVE STATUS
055712  042716  147677                  BIC     #^C<PIP!MOL!VV>,(SP) ;CLEAR DONT CARES
055716  022726  010100                  CMP     #MOL!VV,(SP)+    ;IS DRIVE STATUS OK??
055722  001522                          BEQ     380$             ;YES!!

                                ;REPORT ERROR IF POSITIONING IN PROGRESS AND NO SEEK INCOMPLETE ERROR,
                                ;I.E. PIP = 1 AND SKI = 0
055724  032737  020000  001346          BIT     #PIP,RMDSI       ;IS "PIP" SET??
055732  001430                          BEQ     360$             ;NO!
055734  032737  040000  001376          BIT     #SKI,RMER2I      ;WAS "SKI" ERROR REPORTED??
```

```
                                        BNE     360$            ;YES-DONT REPORT PIP
056742  001024
056744  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056752  042737  020000  001140        BIC     #PIP,$GDDAT
056760  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056766  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
056772  112776  000333  000000        MOVB    #333,@(SP)      ;WRITE ERROR NUMBER
056000  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056004  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056006  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
056012  000240                        NOP
056014                          360$:

                                ;REPORT ERROR IF MEDIUM IS NOT ON LINE AND OPI ERROR WAS NOT
                                ;REPORTED, I.E., MOL = OPI = 0
056014  032737  010000  001346        BIT     #MOL,RMDSI      ;IS MEDIUM ON LINE??
056022  001027                        BNE     370$            ;YES!!
056024  032737  020000  001350        BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
056032  001023                        BNE     370$            ;YES!!
056034  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056042  052737  010000  001140        BIS     #MOL,$GDDAT
056050  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056056  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
056062  112776  000334  000000        MOVB    #334,@(SP)      ;WRITE ERROR NUMBER
056070  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056074  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056076  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
056102                          370$:

                                ;REPORT ERROR IF VOLUME IS NOT VALID AND "IVC" ERROR WAS NOT
                                ;REPORTED, I.E., VV = IVC = 0
056102  032737  000100  001346        BIT     #VV,RMDSI       ;IS VOLUME VALID??
056110  001027                        BNE     380$            ;YES!!
056112  032737  010000  001376        BIT     #IVC,RMER2I     ;WAS IVC ERROR REPORTED??
056120  001033                        BNE     390$            ;YES!!
056122  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056130  052737  000100  001140        BIS     #VV,$GDDAT
056136  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056144  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
056150  112776  000335  000000        MOVB    #335,@(SP)      ;WRITE ERROR NUMBER
056156  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056162  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056164  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
056170                          380$:

                                ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS FOUND
056170  062716  000004                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
056174  105776  000000                TSTB    @(SP)           ;ANY ERROR??
056200  001403                        BEQ     390$            ;NO!!
056202  062716  000004                ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
056206  000402                        BR      400$
056210  162716  000004          390$: SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN

056214  000207                  400$: RTS     PC              ;RETURN TO USER

056216  000000                  500$: .WORD                   ;ERROR FLAGS
056220  000000                  510$: .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
                                
056222                          STCDRVSTS:
                                ;CLEAR USER'S ERROR CALL
056222  062716  000004                ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
056226  105076  000000                CLRB    @(SP)           ;CLEAR ERROR NUMBER
056232  162716  000004                SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR RETURN
                                ;SEE IF "MOL" = "VV" = 1, AND "PIP" = 0
056236  013746  001346                MOV     RMDSI,-(SP)     ;PUT DRIVE STATUS ON STACK
056242  042716  147677                BIC     #^C<PIP!MOL!VV>,(SP)
056246  022726  010100                CMP     #MOL!VV,(SP)+   ;ARE MOL,VV AND PIP O.K.??
056252  001524                      BEQ     30$             ;YES!!
                                
                                ;REPORT AN ERROR IF MOL = 0 AND "OPI" = 0
056254  032737  010000  001346        BIT     #MOL,RMDSI      ;IS MOL ON ??
056262  001030                      BNE     10$             ;YES!!
056264  032737  020000  001350        BIT     #OPI,RMER1I     ;WAS "OPI" SET??
056272  001024                      BNE     10$             ;YES-DONT REPORT "MOL"   0
056274  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056302  052737  010000  001140        BIS     #MOL,$GDDAT
056310  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056316  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056322  112776  000207  000000        MOVB    #207,@(SP)      ;WRITE ERROR NUMBER IN CALL
056330  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056334  004736                      JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056336  162716  000010                SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
056342  000240                      NOP
056344                          10$:
                                
                                ;REPORT AN ERROR IF VOLUME VALID IS NOW ZERO AND "IVC" - 0
056344  032737  000100  001346        BIT     #VV,RMDSI       ;IS "VV" = 0??
056352  001030                      BNE     20$             ;NO.!
```

```
 058 056354  032737  010000  001376          BIT     #IVC,RMER2I     ;WAS "IVC" SET??
 059 056362  001024                          BNE     20$             ;YES-DONT REPORT "VV" = 0
 060 056364  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
 061 056372  052737  000100  001346          BIS     #VV,RMDSI
 062 056400  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
 063 056406  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 064 056412  112776  000210  000000          MOVB    #210,@(SP)      ;WRITE ERROR NUMBER IN CALL
 065 056420  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 066 056424  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 067 056426  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 068 056432  000240                          NOP
 069 056434                          20$:
 070
 071                                 ;REPORT AN ERROR IF DRIVE IS OFF CYLINDER AND "SKI" = 0
 072 056434  032737  020000  001346          BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
 073 056442  001430                          BEQ     30$             ;NO!!
 074 056444  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS "SKI" SET??
 075 056452  001024                          BNE     30$             ;YES-DONT REPORT "PIP" = 1
 076 056454  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
 077 056462  042737  020000  001140          BIC     #PIP,$GDDAT
 078 056470  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
 079 056476  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 080 056502  112776  000211  000000          MOVB    #211,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
 081 056510  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 082 056514  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 083 056516  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 084 056522  000240                          NOP
 085 056524                          30$:
 086
 087                                 ;SEE IF "SKI" = "DVC" = 0
 088 056524  013746  001376                  MOV     RMER2I,-(SP)    ;PUT ERROR REG 2 ON STACK
 089 056530  042726  137577                  BIC     #^C<SKI!DVC>,(SP)+
 090 056534  001460                          BEQ     60$             ;BRANCH IF NO ERROR
 091 056536                          40$:
 092
 093                                 ;REPORT AN ERROR IF THERE IS A DEVICE FAULT
 094 056536  032737  000200  001376          BIT     #DVC,RMER2I     ;ANY DEVICE FAULT??
 095 056544  001424                          BEQ     50$             ;NO!!
 096 056546  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
 097 056554  042737  000200  001140          BIC     #DVC,$GDDAT
 098 056562  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
 099 056570  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S CALL
100 056574  112776  000212  000000          MOVB    #212,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
101 056602  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
102 056606  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
103 056610  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
104 056614  000240                          NOP
105 056616                          50$:
106
107                                 ;REPORT AN ERROR IF "SKI" = 1
108 056616  032737  040000  001376          BIT     #SKI,RMER2I     ;IS THERE A SEEK INCOMPLETE ERROR
109 056624  001424                          BEQ     60$             ;NO!!
110 056626  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
111 056634  042737  040000  001140          BIC     #SKI,$GDDAT
112 056642  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
113 056650  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
114 056654  112776  000213  000000          MOVB    #213,@(SP)      ;WRITE ERROR NUMBER IN USER'S ERROR CALL
```

```
115 056662  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
116 056666  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
117 056670  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
118 056674  000240                          NOP
119 056676                          60$:
120
121                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
122 056676  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
123 056702  105776  000000                  TSTB    @(SP)           ;WAS AN ERROR DETECTED??
124 056706  001403                          BEQ     70$             ;NO!!
125 056710  062716  000004                  ADD     #4,(SP)         ;YES - MOVE SP TO USER'S ERROR RETURN
126 056714  000402                          BR      80$
127 056716  162716  000004          70$:    SUB     #4,(SP)         ;NO - MOVE SP TO NO ERROR RETURN
128 056722  000240                  80$:    NOP
129 056724  000207                          RTS     PC              ;RETURN TO USER
130
```

```
                        .SBTTL  STOP AND SHUTDOWN SUBROUTINES

056726                  STOP:

                        ;DROP PRIORITY TO ALLOW CONSOLE INTERRUPT
056726  012746  000140          MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
056732  012746  056740          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
056736  000002                  RTI                     ;;POP NEW PC AND PS
056740                  64$:
056740  000240                  NOP

                        ;RAISE PRIORITY TO INHIBIT CONSOLE INTERRUPT
056742  012746  000300          MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
056746  012746  056754          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
056752  000002                  RTI                     ;;POP NEW PC AND PS
056754                  65$:
056754  000207                  RTS     PC              ;CONTINUE

056756  005737  001326  SHUT:   TST     CTLFG           ;WAS CONTROL C FLAGGED ?
056762  001002                  BNE     5$              ;BR IF YES
056764  000137  007764          JMP     READY           ;CONTINUE
056770  005737  000042  5$:     TST     @#42            ;ANY MONITOR PRESENT ?
056774  001015                  BNE     10$             ;BR IF YES
056776  104401  057004          TYPE    ,65$            ;;TYPE ASCIZ STRING
057002  000410                  BR      64$             ;;GET OVER THE ASCIZ
                        ;;65$:  .ASCIZ  <CRLF><07>/TEST HALTED/<CRLF>
057024                  64$:
057024  000137  005420          JMP     START           ;GO TO START
057030  000137  032340  10$:     JMP     $EOP            ;RETURN CONTROL TO MONITOR

057034  012737  177777  001326  SHUT2:  MOV     #-1,CTLFG       ;SET THE CONTROL-C FLAG
057042  000002                  RTI                     ;EXIT FROM INTERRUPT
```

```
      1                                 .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

    057044                              $SAVREG:
    057044  010046                              MOV     R0,-(SP)        ;;PUSH R0 ON STACK
    057046  010146                              MOV     R1,-(SP)        ;;PUSH R1 ON STACK
    057050  010246                              MOV     R2,-(SP)        ;;PUSH R2 ON STACK
    057052  010346                              MOV     R3,-(SP)        ;;PUSH R3 ON STACK
    057054  010446                              MOV     R4,-(SP)        ;;PUSH R4 ON STACK
    057056  010546                              MOV     R5,-(SP)        ;;PUSH R5 ON STACK
    057060  016646  000022                      MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
    057064  016646  000022                      MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
    057070  016646  000022                      MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
    057074  016646  000022                      MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
    057100  000002                              RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
    057102                              $RESREG:
    057102  012666  000022                      MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
    057106  012666  000022                      MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
    057112  012666  000022                      MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
    057116  012666  000022                      MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
    057122  012605                              MOV     (SP)+,R5        ;;POP STACK INTO R5
    057124  012604                              MOV     (SP)+,R4        ;;POP STACK INTO R4
    057126  012603                              MOV     (SP)+,R3        ;;POP STACK INTO R3
    057130  012602                              MOV     (SP)+,R2        ;;POP STACK INTO R2
    057132  012601                              MOV     (SP)+,R1        ;;POP STACK INTO R1
    057134  012600                              MOV     (SP)+,R0        ;;POP STACK INTO R0
    057136  000002                              RTI
      2                                 .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

    057140  010146                      $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
    057142  016601  000006                      MOV     6(SP),R1        ;;GET THE INPUT NUMBER
    057146  000261                              SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
```

```
057150  112737  000060  057212  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
057156  006101                          ROL     R1              ;;GET THIS BIT
057160  001406                          BEQ     2$              ;;DONE?
057162  105537  057212                  ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
057166  104401  057212                  TYPE    ,$BIN           ;;GO TYPE THIS BIT
057172  000241                          CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
057174  000765                          BR      1$              ;;GO DO THE NEXT BIT
057176  012601                  2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
057200  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
057206  012616                          MOV     (SP)+,(SP)
057210  000002                          RTI                     ;;RETURN TO USER
057212     000     000          $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
      3                         .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

057214                          $TYPDS:
057214  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
057216  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
057220  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
057222  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
057224  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
057226  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
057232  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
057236  100004                          BPL     1$              ;;BR IF INPUT IS POS.
057240  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
057242  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
057250  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
057252  012703  057430                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
057256  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
057262  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
057264  016001  057420                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
057270  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
057272  002402                          BLT     4$              ;;BR IF DONE
057274  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
057276  000774                          BR      3$
057300  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
057302  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
057304  001002                          BNE     5$              ;;FALL THROUGH IF 0
057306  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
057310  100407                          BMI     7$              ;;BR IF YES
057312  106316                  5$:     ASLB    (SP)            ;;MSD?
057314  103003                          BCC     6$              ;;BR IF NO
057316  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
057324  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
057330  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
057334  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
057336  005720                          TST     (R0)+           ;;JUST INCREMENTING
057340  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
```

```
057344  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
057346  003002                          BGT     8$              ;;GO TO EXIT
057350  010502                          MOV     R5,R2           ;;GET THE LSD
057352  000764                          BR      6$              ;;GO CHANGE TO ASCII
057354  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
057356  100003                          BPL     9$              ;;BR IF NO
057360  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
057366  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
057370  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
057372  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
057374  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
057376  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
057400  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
057402  104401  057430                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
057406  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
057414  012616                          MOV     (SP)+,(SP)
057416  000002                          RTI                     ;;RETURN TO USER
057420  023420                  $DTBL:  10000.
057422  001750                          1000.
057424  000144                          100.
057426  000012                          10.
057430                          $DBLK:  .BLKW   4
    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

057440  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
057444  116637  000001  057663        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
057452  112637  057665                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
057456  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
057462  000406                        BR      $TYPON
057464  112737  000001  057663  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
057472  112737  000006  057665        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
057500  112737  000005  057662  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
057506  010346                        MOV     R3,-(SP)        ;;SAVE R3
057510  010446                        MOV     R4,-(SP)        ;;SAVE R4
```

```
057512  010546                          MOV     R5,-(SP)        ;;SAVE R5
057514  113704  057665                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
057520  005404                          NEG     R4
057522  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
057526  110437  057664                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
057532  113704  057663                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
057536  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
057542  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
057544  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
057546  000404                          BR      3$              ;;GO DO MSB
057550  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
057552  006105                          ROL     R5
057554  006105                          ROL     R5
057556  010503                          MOV     R5,R3
057560  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
057562  10533?  057664                  DECB    $OMODE          ;;TYPE THIS DIGIT?
057566  100016                          BPL     7$              ;;BR IF NO
057570  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
057574  001002                          BNE     4$              ;;TEST FOR 0
057576  005704                          TST     R4              ;;SUPPRESS THIS 0?
057600  001403                          BEQ     5$              ;;BR IF YES
057602  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
057604  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
057610  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
057614  110337  057660                  MOVB    R3,8$           ;;SAVE FOR TYPING
057620  104401  057660                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
057624  10533?  057662          7$:     DECB    $OCNT           ;;COUNT BY 1
057630  003347                          BGT     2$              ;;BR IF MORE TO DO
057632  002402                          BLT     6$              ;;BR IF DONE
057634  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
057636  000744                          BR      2$              ;;GO DO THE LAST DIGIT
057640  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
057642  012604                          MOV     (SP)+,R4        ;;RESTORE R4
057644  012603                          MOV     (SP)+,R3        ;;RESTORE R3
057646  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
057654  012616                          MOV     (SP)+,(SP)
057656  000002                          RTI                     ;;RETURN
057660     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
057661     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
057662     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
057663     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
057664  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5                               .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
```

```
                                ;*
057666  105737  001173        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
057672  100002                        BPL     1$              ;;BR IF YES
057674  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
057676  000430                        BR      3$              ;;LEAVE
057700  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
057702  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
057706  122737  000001  001242        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
057714  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
057716  132737  000100  001243        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
057724  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
057726  010037  057736                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
057732  004737  062660                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
057736  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
057740  132737  000040  001243  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
057746  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
057750  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
057752  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
057754  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
057756  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
057760  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
057764  000002                        RTI                     ;;RETURN
057766  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
057772  001430                        BEQ     8$
057774  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
060000  001006                        BNE     5$
060002  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
060004  104401                        TYPE                    ;;TYPE A CR AND LF
060006  001217                        $CRLF
060010  105037  060216                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
060014  000755                        BR      2$              ;;GET NEXT CHARACTER
060016  004737  060100          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
060022  123726  001172          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
060026  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
060030  013746  001170                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                              ;;AND THE NULL CHAR.
060034  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
060040  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
060042  004737  060100                JSR     PC,$TYPEC       ;;GO TYPE A NULL
060046  105337  060216                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
060052  000770                        BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

060054  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
060060  004737  060100          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
060064  132737  000007  060216        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
060072  001372                        BNE     9$              ;;TAB STOP
060074  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
060076  000724                        BR      2$              ;;GET NEXT CHARACTER
060100                          $TYPEC:
060100  105777  121054                TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
060104  100022                        BPL     10$             ;;BR IF NOT
060106  017746  121050                MOV     @$TKB,-(SP)     ;;GET CHAR
060112  042716  177600                BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
060116  122716  000023                CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
```

```
060122  001012                          BNE     102$            ;;BR IF NOT
060124                          101$:
060124  105777  121030                  TSTB    @$TKS           ;;WAIT FOR CHAR
060130  100375                          BPL     101$
060132  117716  121024                  MOVB    @$TKB,(SP)      ;;GET CHAR
060136  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
060142  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
060146  001366                          BNE     101$            ;;BR IF NOT
060150                          102$:
060150  005726                          TST     (SP)+           ;;FIX STACK
060152                          10$:
060152  105777  121006                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
060156  100375                          BPL     10$
060160  116677  000002  121000          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
060166  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
060174  001003                          BNE     1$              ;;BRANCH IF NO
060176  105037  060216                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
060202  000406                          BR      $TYPEX          ;;EXIT
060204  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
060212  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
060214  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
060216  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
060220  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

060222                  $SCOPE:
060222  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
060224  004737  056726                  JSR     PC,STOP
060230  032777  040000  120716  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
060236  001131                          BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
060240  000416          $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP-240)
060242  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
060246  012737  060266  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
060254  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
060260  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
060264  000500                          BR      $SVLAD          ;;GO TO THE NEXT TEST
060266  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
060270  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
060274  000440                          BR      7$              ;;LOOP ON THE PRESENT TEST
060276                  6$:;#####END OF CODE FOR THE XOR TESTER#####
060276  032777  000400  120650          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
060304  001421                          BEQ     2$              ;;BR IF NO
```

```
060306  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
060310  117716  120640                  MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
060314  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
060316  022716  000033                  CMP     #33,(SP)        ;;CHECK THE NUMBER IN THE SWR
060322  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
060324  011637  001116                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
060330  005316                          DEC     (SP)            ;;BACKUP BY ONE
060332  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
060334  062716  060540                  ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
060340  013637  001122                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
060344  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
060346  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
060350  105737  001117          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
060354  001421                          BEQ     3$              ;;BR IF NO
060356  123737  001131  001117          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
060364  101015                          BHI     3$              ;;BR IF NO
060366  032777  001000  120560          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
060374  001404                          BEQ     4$              ;;BR IF NO
060376  013737  001124  001122  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
060404  000446                          BR      $OVER
060406  105037  001117          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
060412  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
060416  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
060420  032777  004000  120526  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
060426  001011                          BNE     1$              ;;BR IF YES
060430  005737  001230                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
060434  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
060436  005237  001120                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
060442  023737  001206  001120          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
060450  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
060452  012737  000001  001120  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
060460  013737  060536  001206          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
060466  105237  001116          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
060472  113737  001116  001226          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
060500  011637  001122                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
060504  011637  001124                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
060510  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
060514  112737  000001  001131          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
060522  013777  001116  120426  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
060530  013716  001122                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
060534  000002                          RTI                     ;;FIXES PS
060536  000012                  $MXCNT: 10.                     ;;MAX. NUMBER OF ITERATIONS
060540                          $SW08TBL:
060540  010132                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
060542  010316                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
060544  010502                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
060546  010704                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
060550  011546                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
060552  012402                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
060554  013172                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
060556  014132                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
060560  014752                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
060562  015542                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
060564  016500                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
060566  017274                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
060570  020106                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
060572  020720                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
```

```
060574  021542                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
060576  022304                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
060600  023046                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
060602  023610                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
060604  024114                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
060606  024420                          .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
060610  024762                          .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
060612  025304                          .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
060614  025622                          .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
060616  026534                          .WORD   TST30+2         ;;STARTING ADDRESS OF TEST 30
060620  027356                          .WORD   TST31+2         ;;STARTING ADDRESS OF TEST 31
060622  030140                          .WORD   TST32+2         ;;STARTING ADDRESS OF TEST 32
060624  031310                          .WORD   TST33+2         ;;STARTING ADDRESS OF TEST 33
      7                         .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
060626                          $ERROR:
060626  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
060630  105237  001117          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
060634  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
060636  013777  001116  120312          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
060644  032777  002000  120302          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
060652  001402                          BEQ     1$              ;;NO - SKIP
060654  104401  001212                  TYPE    ,$BELL          ;;RING BELL
060660  005237  001126          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
060664  011637  001132                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
060670  162737  000002  001132          SUB     #2,$ERRPC
060676  117737  120230  001130          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
060704  032777  020000  120242          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
060712  001004                          BNE     20$             ;;SKIP TYPEOUTS
060714  004737  032550                  JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
060720  104401  001217                  TYPE    ,$CRLF
060724                          20$:
060724  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
060732  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
060734  113737  001130  060746          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
060742  004737  062670                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
060746     000                  21$:    .BYTE   0
060747     000                          .BYTE   0
060750  000777                  22$:    BR      22$             ;;APT ERROR LOOP
060752  005777  120176          2$:     TST     @SWR            ;;HALT ON ERROR
060756  100002                          BPL     3$              ;;SKIP IF CONTINUE
060760  000000                          HALT                    ;;HALT ON ERROR!
060762  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
060764  032777  001000  120162  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
060772  001402                          BEQ     4$              ;;BR IF NO
```

```
060774  013716  001124                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
061000  005737  001210          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
061004  001402                          BEQ     5$              ;;BR IF NONE
061006  013716  001210                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
061012                          5$:
061012  022737  032530  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
061020  001001                          BNE     6$              ;;BRANCH IF NO
061022  000000                          HALT                    ;;YES
061024                          6$:
061024  000002                          RTI                     ;;RETURN
      8                         .SBTTL  TTY INPUT ROUTINE

                                ;;**********************************************************************
                                .ENABL  LSB
061026  000000                $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
061030  000000                $TKQIN: .WORD   0               ;;INPUT POINTER
061032  000000                $TKQOUT: .WORD  0               ;;OUTPUT POINTER
061034                          $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        061035                  $TKQEND=.
                                .EVEN

                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN
                                ;
061036  005037  061026          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
061042  012737  061034  061030          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
061050  013737  061030  061032          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
061056  012737  061106  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
061064  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
061072  005777  120064                  TST     @$TKB           ;;CLEAR DONE FLAG
061076  012777  000100  120054          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
061104  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)
                                ;
061106  117746  120050          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
061112  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
061116  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
061122  001007                          BNE     1$              ;;BRANCH IF NO
061124  104401  062222                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
061130  004737  061036                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
061134  005726                          TST     (SP)+           ;;CLEAN UP STACK
061136  000137  057034                  JMP     SHUT2           ;;CONTROL C RESTART
061142  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
061146  001004                          BNE     2$              ;;BRANCH IF NO
061150  022737  000176  001154          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
061156  001500                          BEQ     6$              ;;GO TO SWR CHANGE
```

```
061160                          2$:
061160  022737  000001  061026          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
061166  001004                          BNE     3$              ;;BRANCH IF NO
061170  104401  001212                  TYPE    ,$BELL          ;;RING THE TTY BELL
061174  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
061176  000451                          BR      5$              ;;EXIT
061200  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
061204  001021                          BNE     32$             ;;BRANCH IF NO
061206  005077  117746                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
061212  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
061214  105777  117740          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
061220  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
061222  117746  117734                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
061226  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
061232  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
061236  001366                          BNE     31$             ;;BRANCH IF NO
061240  012777  000100  117712          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
061246  000002                          RTI                     ;;RETURN
061250  005237  061026          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
061254  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
061260  002405                          BLT     4$              ;;BRANCH IF YES
061262  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
061266  003002                          BGT     4$              ;;BRANCH IF YES
061270  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
061274  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
061300  005237  061030                  INC     $TKQIN          ;;UPDATE THE POINTER
061304  023727  061030  061035          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
061312  001003                          BNE     5$              ;;BRANCH IF NO
061314  012737  061034  061030          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
061322  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.
061324  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
061332  001124                          BNE     15$             ;;EXIT IF NOT
061334  105777  117620                  TSTB    @$TKS           ;;IS A CHAR WAITING?
061340  100121                          BPL     15$             ;;IF NOT, EXIT
061342  117746  117614                  MOVB    @$TKB,-(SP)     ;;YES
061346  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
061352  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
061356  001300                          BNE     2$              ;;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.
061360  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
061366  001674                          BEQ     2$              ;;BRANCH IF YES
061370  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
061372  004737  061036                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
061376  005077  117556                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
061402  112737  000001  001151          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
```

```
061410  104401  062234                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
061414  104401  062241          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
061420  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
061424  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
061426  104401  062252                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
061432  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
061434  005046                          CLR     -(SP)           ;;THE NEW SWR
061436  105777  117516          7$:     TSTB    @$TKS           ;;CHAR THERE?
061442  100375                          BPL     7$              ;;IF NOT TRY AGAIN

061444  117746  117512                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
061450  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

061454  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
061460  001015                          BNE     9$              ;;BRANCH IF NOT
061462  104401  062222                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
061466  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
061472  123727  001151  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
061500  001003                          BNE     8$              ;;BRANCH IF NO
061502  012777  000100  117450          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
061510  000137  057034          8$:     JMP     SHUT2           ;;CONTROL-C RESTART


061514  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
061520  001005                          BNE     10$             ;;BRANCH IF NOT
061522  104401  062227                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
061526  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
061532  000737                          BR      19$             ;;LET'S TRY IT AGAIN

061534  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
061540  001022                          BNE     16$             ;;BRANCH IF NO
061542  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
061546  001403                          BEQ     11$             ;;BRANCH IF YES
061550  016677  000002  117376          MOV     2(SP),@SWR      ;;SAVE NEW SWR
061556  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
061562  104401  001217          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
061566  123727  001151  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
061574  001003                          BNE     15$             ;;BRANCH IF NOT
061576  012777  000100  117354          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
061604  000002                  15$:    RTI                     ;;RETURN
061606  004737  060100          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
061612  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
061616  002420                          BLT     18$             ;;BRANCH IF YES
061620  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
061624  003015                          BGT     18$             ;;BRANCH IF YES
061626  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
061632  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
061636  001403                          BEQ     17$             ;;BRANCH IF YES
061640  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
061642  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
061644  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
061646  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
061652  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
061656  000667                          BR      7$              ;;GET THE NEXT ONE
061660  104401  001216          18$.    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
```

```
061664  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
                                ;

061666  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
061670  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
061676  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
061702  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
061704  012746  061712                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
061710  000002                          RTI                     ;;POP NEW PC AND PS
061712                          64$:
061712  005737  061026          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
061716  001775                          BEQ     1$
061720  005337  061026                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
061724  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
061732  005237  061032                  INC     $TKQOUT         ;;UPDATE THE POINTER
061736  023727  061032  061035          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
061744  001003                          BNE     2$              ;;BRANCH IF NO
061746  012737  061034  061032          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
061754  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

061756  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
061760  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
061762  012703  062212          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
061766  022703  062222          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
061772  101456                          BLOS    4$              ;;BR IF YES
061774  104411                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
061776  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
062000  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
062004  001022                          BNE     5$              ;;BR IF NO
062006  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
062010  001007                          BNE     6$              ;;BR IF NO
062012  112737  000134  062210          MOVB    #'\,9$          ;;TYPE A BACK SLASH
062020  104401  062210                  TYPE    ,9$
062024  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
062030  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
062032  020327  062212                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
062036  103434                          BLO     4$              ;;BR IF YES
062040  111337  062210                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
062044  104401  062210                  TYPE    ,9$             ;;GO TYPE
062050  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
062052  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
062054  001406                          BEQ     7$              ;;BR IF NO
```

```
062056  112737  000134  062210          MOVB    #'\,9$          ;;TYPE A BACK SLASH
062064  104401  062210                  TYPE    ,9$
062070  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
062072  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
062076  001003                          BNE     8$              ;;BR IF NO
062100  104401  062227                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
062104  000726                          BR      1$              ;;GO START OVER
062106  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
062112  001011                          BNE     3$              ;;BRANCH IF NO
062114  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
062116  104401  001217                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
062122  104401  062212                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
062126  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
062130  104401  001216          4$:     TYPE    ,$QUES          ;;TYPE A '?'
062134  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
062136  111337  062210          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
062142  104401  062210                  TYPE    ,9$
062146  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
062152  001305                          BNE     2$              ;;LOOP IF NOT RETURN
062154  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
062160  104401  001220                  TYPE    ,$LF            ;;TYPE A LINE FEED
062164  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
062166  012603                          MOV     (SP)+,R3        ;;RESTORE R3
062170  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
062172  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
062200  012766  062212  000004          MOV     #$TTYIN,4(SP)
062206  000002                          RTI                     ;;RETURN
062210     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
062211     000                          .BYTE   0               ;;TERMINATOR
062212                          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
062222     136     103     015  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
062227     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
062234     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
062241     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
062252     040     040     116  $MNEW:  .ASCIZ  /  NEW = /
                                .EVEN
9                               .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

062264  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
062266  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
062274  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
062276  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
062300  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
062302  104412                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
062304  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
062306  005001                          CLR     R1              ;;CLEAR DATA WORD
062310  005002                          CLR     R2
062312  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
062314  001412                          BEQ     3$              ;;IF ZERO GET OUT
```

```
062316  006301                          ASL     R1              ;;*2
062320  006102                          ROL     R2
062322  006301                          ASL     R1              ;;*4
062324  006102                          ROL     R2
062326  006301                          ASL     R1              ;;*8
062330  006102                          ROL     R2
062332  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
062336  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
062340  000764                          BR      2$              ;;LOOP
062342  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
062344  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
062350  010237  062364                  MOV     R2,$HIOCT
062354  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
062356  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
062360  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
062362  000002                          RTI                     ;;RETURN
062364  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
10                                .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.

062366  016646  000002          $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
062372  042716  000020                  BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
062376  012746  062404                  MOV     #1$,-(SP)       ;; T-BIT TRAPS
062402  000002                          RTI                     ;;SET THE NEW STATUS
062404  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
062406  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
062412  005740                          TST     -(R0)           ;;BACKUP BY 2
062414  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
062416  006300                          ASL     R0              ;;POSITION FOR INDEXING
062420  016000  062440                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
062424  000200                          RTS     R0              ;;GO TO ROUTINE


                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

062426  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
062430  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
062436  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
062440  062426                  $TRPAD: .WORD   $TRAP2
062442  057666                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
062444  057464                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
062446  057440                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
062450  057500                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
062452  057214                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
```

```
        062454  057140                          $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

        062456  061414                          $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING

        062460  061324                          $CKSWR  ;;CALL=CKSWR    TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
        062462  061666                          $RDCHR  ;;CALL=RDCHR    TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
        062464  061756                          $RDLIN  ;;CALL=RDLIN    TRAP+12(104412) TTY TYPEIN STRING ROUTINE
        062466  062264                          $RDOCT  ;;CALL=RDOCT    TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
        062470  057044                          $SAVREG ;;CALL=SAVREG   TRAP+14(104414) SAVE R0-R5 ROUTINE
        062472  057102                          $RESREG ;;CALL=RESREG   TRAP+15(104415) RESTORE R0-R5 ROUTINE
    11                          .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**************************************************************
                                ;POWER DOWN ROUTINE
        062474  012737  062634  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
        062502  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
        062510  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
        062512  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
        062514  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
        062516  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
        062520  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
        062522  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
        062524  017746  116424                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
        062530  010637  062640                  MOV     SP,$SAVR6       ;;SAVE SP
        062534  012737  062546  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
        062542  000000                          HALT
        062544  000776                          BR      .-2             ;;HANG UP

                                ;;**************************************************************
                                ;POWER UP ROUTINE
        062546  012737  062634  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
        062554  013706  062640                  MOV     $SAVR6,SP       ;;GET SP
        062560  005037  062640                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
        062564  005237  062640          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
        062570  001375                          BNE     1$              ;;OF  WORD
        062572  012677  116356                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
        062576  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
        062600  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        062602  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        062604  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        062606  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        062610  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
        062612  012737  062474  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
        062620  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
        062626  104401                          TYPE                    ;REPORT THE POWER FAILURE
        062630  062642          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
        062632  000002                          RTI
        062634  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
        062636  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
        062640  000000          $SAVR6: 0                               ;;PUT THE SP HERE
        062642     015     012     120  $POWER: .ASCIZ  <15><12>"POWER"
                                        .EVEN
    12                          .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;**************************************************************
        062652  112737  000001  063116  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
        062660  112737  000001  063114  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
```

```
062666  000403                          BR      $ATYC
062670  112737  000001  063116  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
062676                          $ATYC:
062676  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
062700  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
062702  105737  063114                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
062706  001450                          BEQ     5$              ;;IF NOT:  BR
062710  122737  000001  001242          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
062716  001031                          BNE     3$              ;;IF NOT:  BR
062720  132737  000100  001243          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
062726  001425                          BEQ     3$              ;;IF NOT:  BR
062730  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
062734  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
062742  005737  001222          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
062746  001375                          BNE     1$              ;;IF NOT:  WAIT
062750  010037  001236                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
062754  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
062756  001376                          BNE     2$
062760  163700  001236                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
062764  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
062766  010037  001240                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
062772  012737  000004  001222          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
063000  000413                          BR      5$
063002  017637  000004  063026  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
063010  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
063016  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
063022  004737  057666                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
063026  000000                  4$:     .WORD   0
063030                          5$:
063030  105737  063116          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
063034  001416                          BEQ     12$             ;;IF NOT:  BR
063036  005737  001242                  TST     $ENV            ;;RUNNING UNDER APT?
063042  001413                          BEQ     12$             ;;IF NOT:  BR
063044  005737  001222          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
063050  001375                          BNE     11$             ;;IF NOT:  WAIT
063052  017637  000004  001224          MOV     @4(SP),$FATAL   ;;GET ERROR #
063060  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
063066  005237  001222                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
063072  105037  063116          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
063076  105037  063115                  CLRB    $LFLG           ;;CLEAR LOG FLAG
063102  105037  063114                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
063106  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
063110  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
063112  000207                          RTS     PC              ;;RETURN
063114     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
063115     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
063116     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
                                        .EVEN
            000200              APTSIZE = 200
            000001              APTENV  = 001
            000100              APTSPOOL= 100
            000040              APTCSUP = 040
```

```
                                        .SBTTL  CONSOLE MESSAGES
                                
063120  200     106     101     SCTMSG: .ASCII  <CRLF>@FAILED TO RECOVER THE BAD SECTOR FILE(DEC 144)@
063177  200     117     116             .ASCIZ  <CRLF>@ON THIS DRIVE@
063216  075     000             EQUALS: .ASCIZ  @=@
063220  101     114     114     ALL:    .ASCIZ  @ALL@<CRLF>
063225  040     077     040     QUES:   .ASCIZ  @ ? @
063231  054     040     000     COMMA:  .ASCIZ  @, @
063234  200     124     117     MSHELP: .ASCII  <CRLF>@TO  ENSURE THAT NO  BAD HEADERS ARE LEFT  ON THE DISK@
063322  200     120     101             .ASCII  <CRLF>@PACK,  THIS PROGRAM SHOULD BE HALTED BY TYPING A (^C)@
063410  200     103     117             .ASCII  <CRLF>@CONTROL C.  AS A RESULT,  THE PROGRAM WILL  BE HALTED@
063476  200     127     110             .ASCII  <CRLF>@WHEN THE DRIVE UNDER TEST HAS COMPLETED TESTING.@<CRLF>
063560  200     124     131             .ASCIZ  <CRLF>@TYPE HELP TEXT (Y/N) ? @
063611                          UBUSQST:
063611  200     103     110             .ASCIZ  <CRLF>@CHANGE ADDRESSES (Y/N) ? @
063644  200     125     123     CNSL00: .ASCIZ  <CRLF>@USE SAME DEVICES (Y/N) ? @
063677  200     102     125     CNSL01: .ASCIZ  <CRLF>@BUS ADDRESS @
063715  040     114     111     CNSL02: .ASCIZ  @ LIMITS - LO= 160000, HI= 17XXXX@<CRLF>
063757  126     105     103     CNSL03: .ASCIZ  @VECTOR ADDRESS @
063777  040     114     111     CNSL04: .ASCIZ  @ LIMITS - LO= 0, HI= 1000@<CRLF><LF>
064033  102     122     04C     CNSL05: .ASCIZ  @BR LEVEL @
064045  040     114     111     CNSL06: .ASCIZ  @ LIMITS - LO= 0, HI= 7@<CRLF><LF>
064076  200                     CNSL07: .ASCII  <CRLF>
064077  200     124     131             .ASCII  <CRLF>@TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S)@
064164  200     101     116             .ASCIZ  <CRLF>@AND TERMINATE INPUT WITH A CARRIAGE RETURN.@
064241  200                     CNSL08: .ASCII  <CRLF>
064242  040     077     111     CNSL09: .ASCIZ  @ ?ILLEGAL INPUT@<CRLF>
064263  200     104     122     MSDRVS: .ASCIZ  <CRLF>/DRIVE(S): /
064277  104     122     111     MSGDRV: .ASCIZ  /DRIVE/
064305  040     111     123     LODEV:  .ASCIZ  / IS LOAD DEVICE/
064325  040     116     117     NOTPRS: .ASCIZ  / NOT PRESENT/
064342  040     116     117     NOTAVL: .ASCIZ  / NOT AVAILABLE/
064361  040     117     106     OFFLIN: .ASCIZ  / OFF LINE/
064373  040     117     116     ONLINE: .ASCIZ  / ON LINE /

                                        .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 IS 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.

064406                  FNCDTB:                                 ;FUNCTION CODE TABLE

064406  020000          .WORD   OPI                             ;NOP
```

```
 58 064410  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (2)
 59 064412  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEEK
 60 064414  130000                .WORD   ATA!OPI!IVC                     ;RECALIBRATE
 61 064416  020000                .WORD   OPI                             ;DRIVE CLEAR
 62 064420  030000                .WORD   OPI!IVC                         ;RELEASE
 63 064422  130000                .WORD   OPI!ATA!IVC                     ;OFFSET
 64 064424  130000                .WORD   OPI!ATA!IVC                     ;RETURN TO CENTERLINE
 65 064426  020000                .WORD   OPI                             ;READ IN PRESET
 66 064430  020000                .WORD   OPI                             ;PACK ACKNOWLEDGE
 67 064432  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (24)
 68 064434  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (26)
 69 064436  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEARCH
 70 064440  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (32)
 71 064442  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (34)
 72 064444  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (36)
 73 064446  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (40)
 74 064450  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (42)
 75 064452  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (44)
 76 064454  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (46)
 77 064456  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
 78 064460  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK HEADER AND DATA
 79 064462  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (54)
 80 064464  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (56)
 81 064466  037200                .WORD   OPI!IVC!WLE!IAE.AOE!HCE         ;WRITE DATA
 82 064470  037000                .WORD   OPI!IVC!WLE!IAE!AOE             ;WRITE HEADER AND DATA
 83 064472  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (64)
 84 064474  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (66)
 85 064476  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ DATA
 86 064500  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ HEADER AND DATA
 87 064502  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (74)
 88 064504  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (76)
 89
```

```
                                .SBTTL  ATTENTION (ATA) TABLE

064506      001         ATNTBL: .BYTE   1.
064507      002                 .BYTE   2.
064510      004                 .BYTE   4.
064511      010                 .BYTE   8.
064512      020                 .BYTE   16.
064513      040                 .BYTE   32.
064514      100                 .BYTE   64.
064515      200                 .BYTE   128.
```

```
                                .SBTTL  DATA PATTERN TABLE

064516                          RGDTPT:
064516                          MIXED:
064516  000000                          .WORD   0.
064520  000001                          .WORD   1.
064522  000003                          .WORD   3.
064524  000007                          .WORD   7.
064526  000017                          .WORD   15.
064530  000037                          .WORD   31.
064532  000077                          .WORD   63.
064534  000177                          .WORD   127.
064536  000377                          .WORD   255.
064540  000777                          .WORD   511.
064542  001777                          .WORD   1023.
064544  003777                          .WORD   2047.
064546  007777                          .WORD   4095.
064550  017777                          .WORD   8191.
064552  037777                          .WORD   16383.
064554  077777                          .WORD   32767.
064556  177777                  ONES:   .WORD   65535.
064560  177777                          .WORD   65535.
064562  077777                          .WORD   32767.
064564  037777                          .WORD   16383.
064566  017777                          .WORD   8191.
064570  007777                          .WORD   4095.
064572  003777                          .WORD   2047.
064574  001777                          .WORD   1023.
064576  000777                          .WORD   511.
064600  000377                          .WORD   255.
064602  000177                          .WORD   127.
064604  000077                          .WORD   63.
064606  000037                          .WORD   31.
064610  000017                          .WORD   15.
064612  000007                          .WORD   7.
064614  000003                          .WORD   3.
064616  000001                          .WORD   1.
064620  000000                  ZEROS:  .WORD   0.
064622  000000                          .WORD   0.
064624  000001                          .WORD   1.
064626  000002                          .WORD   2.
064630  000004                          .WORD   4.
064632  000010                          .WORD   8.
064634  000020                          .WORD   16.
064636  000040                          .WORD   32.
064640  000100                          .WORD   64.
064642  000200                          .WORD   128.
064644  000400                          .WORD   256.
064646  001000                          .WORD   512.
064650  002000                          .WORD   1024.
064652  004000                          .WORD   2048.
064654  010000                          .WORD   4096.
064656  020000                          .WORD   8192.
064660  040000                          .WORD   16384.
064662  100000                          .WORD   32768.
064664  100000                          .WORD   32768.
064666  040000                          .WORD   16384.
```

```
 58 064670  020000                    .WORD   8192.
 59 064672  010000                    .WORD   4096.
 60 064674  004000                    .WORD   2048.
 61 064676  002000                    .WORD   1024.
 62 064700  001000                    .WORD   512.
 63 064702  000400                    .WORD   256.
 64 064704  000200                    .WORD   128.
 65 064706  000100                    .WORD   64.
 66 064710  000040                    .WORD   32.
 67 064712  000020                    .WORD   16.
 68 064714  000010                    .WORD   8.
 69 064716  000004                    .WORD   4.
 70 064720  000002                    .WORD   2.
 71 064722  000001                    .WORD   1.
 72 064724  000000                    .WORD   0.
 73 064726  177777                    .WORD   65535.
 74 064730  177776                    .WORD   65534.
 75 064732  177774                    .WORD   65532.
 76 064734  177770                    .WORD   65528.
 77 064736  177760                    .WORD   65520.
 78 064740  177740                    .WORD   65504.
 79 064742  177700                    .WORD   65472.
 80 064744  177600                    .WORD   65408.
 81 064746  177400                    .WORD   65280.
 82 064750  177000                    .WORD   65024.
 83 064752  176000                    .WORD   64512.
 84 064754  174000                    .WORD   63488.
 85 064756  170000                    .WORD   61440.
 86 064760  160000                    .WORD   57344.
 87 064762  140000                    .WORD   49152.
 88 064764  100000                    .WORD   32768.
 89 064766  000000                    .WORD   0.
 90 064770  000000                    .WORD   0.
 91 064772  100000                    .WORD   32768.
 92 064774  140000                    .WORD   49152.
 93 064776  160000                    .WORD   57344.
 94 065000  170000                    .WORD   61440.
 95 065002  174000                    .WORD   63488.
 96 065004  176000                    .WORD   64512.
 97 065006  177000                    .WORD   65024.
 98 065010  177400                    .WORD   65280.
 99 065012  177600                    .WORD   65408.
100 065014  177700                    .WORD   65472.
101 065016  177740                    .WORD   65504.
102 065020  177760                    .WORD   65520.
103 065022  177770                    .WORD   65528.
104 065024  177774                    .WORD   65532.
105 065026  177776                    .WORD   65534.
106 065030  177777                    .WORD   65535.
107 065032  125252            EARLY:  .WORD   43690.
108 065034  152525                    .WORD   43690./2
109 065036  125252                    .WORD   43690.
110 065040  177777                    .WORD   65535.
111 065042  177776                    .WORD   65534.
112 065044  177775                    .WORD   65533.
113 065046  177773                    .WORD   65531.
114 065050  177767                    .WORD   65527.
```

```
115 065052  177757                          .WORD   65519.
116 065054  177737                          .WORD   65503.
117 065056  177677                          .WORD   65471.
118 065060  177577                          .WORD   65407.
119 065062  177377                          .WORD   65279.
120 065064  176777                          .WORD   65023.
121 065066  175777                          .WORD   64511.
122 065070  173777                          .WORD   63487.
123 065072  167777                          .WORD   61439.
124 065074  157777                          .WORD   57343.
125 065076  137777                          .WORD   49151.
126 065100  077777                          .WORD   32767.
127 065102  077777                          .WORD   32767.
128 065104  137777                          .WORD   49151.
129 065106  157777                          .WORD   57343.
130 065110  167777                          .WORD   61439.
131 065112  173777                          .WORD   63487.
132 065114  175777                          .WORD   64511.
133 065116  176777                          .WORD   65023.
134 065120  177377                          .WORD   65279.
135 065122  177577                          .WORD   65407.
136 065124  177677                          .WORD   65471.
137 065126  177737                          .WORD   65503.
138 065130  177757                          .WORD   65519.
139 065132  177767                          .WORD   65527.
140 065134  177773                          .WORD   65531.
141 065136  177775                          .WORD   65533.
142 065140  177776                          .WORD   65534.
143 065142  177777                          .WORD   65535.
144 065144                          ENRGDT:
145
```

```
                                        .SBTTL  ERROR MESSAGE TABLE
                                
 065144  071540  000000          EMT1:   .WORD   EMS1,0
 065150  071607  071624  000000  EMT2:   .WORD   EMS2,EMS3,0
 065156  071607  071667  000000  EMT3:   .WORD   EMS2,EMS4,0
 065164  071732  071762  000000  EMT4:   .WORD   EMS5,EMS6,0
 065172  071732  072074  000000  EMT5:   .WORD   EMS5,EMS10,0
 065200  076535  073753  000000  EMT6:   .WORD   EMS167,EMS64,0
 065206  074521  076562  000000  EMT7:   .WORD   EMS110,EMS170,0
 065214  072027  000000          EMT10:  .WORD   EMS7,0
 065220  072074  000000          EMT11:  .WORD   EMS10,0
 065224  072136  072147  000000  EMT12:  .WORD   EMS11,EMS12,0
 065232  072210  072221  072232  EMT13:  .WORD   EMS13,EMS14,EMS15,EMS16,0
 065244  072304  073753  000000  EMT14:  .WORD   EMS17,EMS64,0
 065252  072136  072367  000000  EMT15:  .WORD   EMS11,EMS21,0
 065260  072136  072412  072541  EMT16:  .WORD   EMS11,EMS22,EMS27,0
 065270  072136  072426  072552  EMT17:  .WORD   EMS11,EMS23,EMS30,0
 065300  072136  072454  072552  EMT20:  .WORD   EMS11,EMS24,EMS30,0
 065310  072136  072503  072541  EMT21:  .WORD   EMS11,EMS25,EMS27,0
 065320  072136  072520  072541  EMT22:  .WORD   EMS11,EMS26,EMS27,0
 065330  072136  072562  072552  EMT23:  .WORD   EMS11,EMS31,EMS30,0
 065340  072136  072611  072552  EMT24:  .WORD   EMS11,EMS32,EMS30,0
 065350  072136  072640  072552  EMT25:  .WORD   EMS11,EMS33,EMS30,0
 065360  072136  072666  072552  EMT26:  .WORD   EMS11,EMS34,EMS30,0
 065370  072136  072737  072552  EMT27:  .WORD   EMS11,EMS35,EMS30,0
 065400  072136  072766  072552  EMT30:  .WORD   EMS11,EMS36,EMS30,0
 065410  072136  073015  072552  EMT31:  .WORD   EMS11,EMS37,EMS30,0
 065420  072136  073043  072552  EMT32:  .WORD   EMS11,EMS40,EMS30,0
 065430  072136  073072  072552  EMT33:  .WORD   EMS11,EMS41,EMS30,0
 065440  072136  073120  072552  EMT34:  .WORD   EMS11,EMS42,EMS30,0
 065450  072136  073147  072552  EMT35:  .WORD   EMS11,EMS43,EMS30,0
 065460  072136  073176  072552  EMT36:  .WORD   EMS11,EMS44,EMS30,0
 065470  072136  073251  072552  EMT37:  .WORD   EMS11,EMS45,EMS30,0
 065500  074037  072344  000000  EMT40:  .WORD   EMS66,EMS20,0
 065506  074225  075736  074233  EMT41:  .WORD   EMS75,EMS141,EMS76,0
 065516  076027  076037  074161  EMT42:  .WORD   EMS144,EMS145,EMS72,EMS76,0
 065530  073343  073457  074233  EMT43:  .WORD   EMS47,EMS53,EMS76,0
 065540  074264  073457  074233  EMT44:  .WORD   EMS77,EMS53,EMS76,0
 065550  074312  073457  074233  EMT45:  .WORD   EMS100,EMS53,EMS76,0
 065560  074340  073457  074233  EMT46:  .WORD   EMS101,EMS53,EMS76,0
 065570  073771  073753  000000  EMT47:  .WORD   EMS65,EMS64,0
 065576  072210  072232  073725  EMT50:  .WORD   EMS13,EMS15,EMS63,0
 065606  072136  073314  072552  EMT51:  .WORD   EMS11,EMS46,EMS30,EMS67,0
 065620  073343  073457  074107  EMT52:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS140,EMS141,0
 065636  073343  073457  074107  EMT53:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS141,EMS164,0
 065654  073372  073457  074107  EMT54:  .WORD   EMS50,EMS53,EMS67,0
 065664  075764  076001  073457  EMT55:  .WORD   EMS142,EMS143,EMS53,EMS67,0
 065676  073421  074161  074107  EMT56:  .WORD   EMS51,EMS72,EMS67,EMS115,EMS50,EMS70,0
 065714  076535  074171  074107  EMT57:  .WORD   EMS167,EMS73,EMS67,0
 065724  073542  074107  074721  EMT60:  .WORD   EMS56,EMS67,EMS115,EMS150,EMS152,EMS70,0
 065742  074146  073542  074107  EMT61:  .WORD   EMS71,EMS56,EMS67,EMS115,EMS150,EMS152,EMS72,0
 065762  075715  074107  074721  EMT62:  .WORD   EMS140,EMS67,EMS115,EMS47,EMS70,0
 065776  000000                  EMT63:  .WORD
 066000  076122  076165  074134  EMT64:  .WORD   EMS150,EMS152,EMS70,EMS115,EMS56,EMS73,EMS67,0
 066020  073447  077162  077343  EMT65:  .WORD   EMS52,EMS205,EMS214,EMS206,EMS115,EMS51,EMS72,0
 066040  076474  074415  074161  EMT66:  .WORD   EMS165,EMS103,EMS72,EMS124,0
 066052  076474  074415  074161  EMT67:  .WORD   EMS165,EMS103,EMS72,EMS171,0
```

```
 58 066064  073314  072344  076367  EMT70:  .WORD   EMS46,EMS20,EMS163,0
 59 066074  074146  074340  076367  EMT71:  .WORD   EMS71,EMS101,EMS163,0
 60 066104  073343  074161  076367  EMT72:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS140,EMS141,0
 61 066122  073343  074161  076367  EMT73:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS141,EMS72,0
 62 066140  073542  073457  076367  EMT74:  .WORD   EMS56,EMS53,EMS163,0
 63 066150  074146  073542  076367  EMT75:  .WORD   EMS71,EMS56,EMS163,EMS115,EMS150,EMS152,EMS72,0
 64 066170  073372  073457  076367  EMT76:  .WORD   EMS50,EMS53,EMS163,0
 65 066200  075764  076001  073457  EMT77:  .WORD   EMS142,EMS143,EMS53,EMS163,0
 66 066212  074225  075736  076367  EMT100: .WORD   EMS75,EMS141,EMS163,EMS115,EMS47,EMS70,0
 67 066230  074225  076122  076367  EMT101: .WORD   EMS75,EMS150,EMS163,EMS115,EMS56,EMS73,0
 68 066246  076535  074171  076367  EMT102: .WORD   EMS167,EMS73,EMS163,0
 69 066256  076105  076141  074206  EMT103: .WORD   EMS147,EMS151,EMS74,EMS163,0
 70 066270  073421  074206  076367  EMT104: .WORD   EMS51,EMS74,EMS163,EMS115,EMS50,EMS70,0
 71 066306  076474  074312  076367  EMT105: .WORD   EMS165,EMS100,EMS163,0
 72 066316  076474  074264  076367  EMT106: .WORD   EMS165,EMS77,EMS163,0
 73 066326  076027  076037  073457  EMT107: .WORD   EMS144,EMS145,EMS53,EMS163,EMS115,EMS143,EMS70,0
 74 066346  074521  074561  074724  EMT110: .WORD   EMS110,EMS112,EMS116,EMS111,0
 75 066360  074645  071667  000000  EMT111: .WORD   EMS113,EMS4,0
 76 066366  074645  071624  000000  EMT112: .WORD   EMS113,EMS3,0
 77 066374  074521  074721  074724  EMT113: .WORD   EMS110,EMS115,EMS116,EMS117,EMS114,0
 78 066410  074645  074776  000000  EMT114: .WORD   EMS113,EMS120,0
 79 066416  075017  075457  075503  EMT115: .WORD   EMS121,EMS132,EMS133,0
 80 066426  075062  075457  075503  EMT116: .WORD   EMS122,EMS132,EMS133,0
 81 066436  075117  075457  075503  EMT117: .WORD   EMS123,EMS132,EMS133,0
 82 066446  075162  075457  075503  EMT120: .WORD   EMS124,EMS132,EMS133,
 83 066456  075214  075457  075503  EMT121: .WORD   EMS125,EMS132,EMS133,0
 84 066466  075257  075457  075503  EMT122: .WORD   EMS126,EMS132,EMS133,0
 85 066476  075660  075457  075503  EMT123: .WORD   EMS137,EMS132,EMS133,0
 86 066506  075361  075457  075503  EMT124: .WORD   EMS130,EMS132,EMS133,0
 87 066516  075417  075457  075503  EMT125: .WORD   EMS131,EMS132,EMS133,0
 88 066526  075017  075457  075526  EMT126: .WORD   EMS121,EMS132,EMS134,EMS123,0
 89 066540  075062  075457  075526  EMT127: .WORD   EMS122,EMS132,EMS134,EMS123,0
 90 066552  075117  075457  075526  EMT130: .WORD   EMS123,EMS132,EMS134,EMS123,0
 91 066564  075162  075457  075526  EMT131: .WORD   EMS124,EMS132,EMS134,EMS123,0
 92 066576  075214  075457  075526  EMT132: .WORD   EMS125,EMS132,EMS134,EMS123,0
 93 066610  075257  075457  075526  EMT133: .WORD   EMS126,EMS132,EMS134,EMS123,0
 94 066622  075660  075457  075526  EMT134: .WORD   EMS137,EMS132,EMS134,EMS123,0
 95 066634  075361  075457  075526  EMT135: .WORD   EMS130,EMS132,EMS134,EMS123,0
 96 066646  075417  075457  075526  EMT136: .WORD   EMS131,EMS132,EMS134,EMS123,0
 97 066660  075017  075457  075570  EMT137: .WORD   EMS121,EMS132,EMS135,0
 98 066670  075117  075457  075570  EMT140: .WORD   EMS123,EMS132,EMS135,0
 99 066700  075017  075457  075633  EMT141: .WORD   EMS121,EMS132,EMS136,0
100 066710  075660  075457  075633  EMT142: .WORD   EMS137,EMS132,EMS136,0
101 066720  075162  075457  075633  EMT143: .WORD   EMS124,EMS132,EMS136,0
102 066730  075214  075457  075633  EMT144: .WORD   EMS125,EMS132,EMS136,0
103 066740  075257  075457  075633  EMT145: .WORD   EMS126,EMS132,EMS136,0
104 066750  075417  075457  075633  EMT146: .WORD   EMS131,EMS132,EMS136,0
105 066760  076611  075457  075633  EMT147: .WORD   EMS171,EMS132,EMS136,0
106 066770  075361  075457  075633  EMT150: .WORD   EMS130,EMS132,EMS136,0
107 067000  075715  074721  075736  EMT151: .WORD   EMS140,EMS115,EMS141,EMS70,0
108 067012  075764  074721  076001  EMT152: .WORD   EMS142,EMS115,EMS143,EMS72,0
109 067024  076027  076037  076066  EMT153: .WORD   EMS144,EMS145,EMS146,EMS115,EMS143,EMS72,0
110 067042  076027  076037  072344  EMT154: .WORD   EMS144,EMS145,EMS20,EMS115,EMS143,EMS70,0
111 067060  076122  076165  076233  EMT155: .WORD   EMS150,EMS152,EMS154,EMS153,0
112 067072  076105  076141  076233  EMT156: .WORD   EMS147,EMS151,EMS154,EMS155,0
113 067104  076105  076141  076267  EMT157: .WORD   EMS147,EMS151,EMS156,EMS157,0
114 067116  076337  076267  076322  EMT160: .WORD   EMS161,EMS156,EMS160,0
```

```
115 067126  072503  072541  076267  EMT161: .WORD   EMS25,EMS27,EMS156,EMS160,0
116 067140  072520  072541  076267  EMT162: .WORD   EMS26,EMS27,EMS156,EMS160,0
117 067152  073343  076367  075715  EMT163: .WORD   EMS47,EMS163,EMS140,0
118 067162  073343  072344  000000  EMT164: .WORD   EMS47,EMS20,0
119 067170  073542  072344  000000  EMT165: .WORD   EMS56,EMS20,0
120 067176  073314  072344  000000  EMT166: .WORD   EMS46,EMS20,0
121 067204  077663  072344  000000  EMT167: .WORD   EMS224,EMS20,0
122 067212  076535  076233  076510  EMT170: .WORD   EMS167,EMS154,EMS166,0
123 067222  076535  076233  076305  EMT171: .WORD   EMS167,EMS154,EMS157,0
124 067232  076535  076233  076247  EMT172: .WORD   EMS167,EMS154,EMS155,0
125 067242  076611  075457  075503  EMT173: .WORD   EMS171,EMS132,EMS133,0
126 067252  076611  075457  075526  EMT174: .WORD   EMS171,EMS132,EMS134,EMS123,0
127 067264  074645  076764  000000  EMT175: .WORD   EMS113,EMS177,0
128 067272  077006  077023  000000  EMT176: .WORD   EMS200,EMS201,0
129 067300  077121  075736  072552  EMT177: .WORD   EMS203,EMS141,EMS30,EMS202,0
130 067312  077121  073421  072552  EMT200: .WORD   EMS203,EMS51,EMS30,EMS202,0
131 067324  077121  077134  072552  EMT201: .WORD   EMS203,EMS204,EMS30,EMS202,0
132 067336  077121  073372  072552  EMT202: .WORD   EMS203,EMS50,EMS30,EMS202,0
133 067350  077121  076001  072552  EMT203: .WORD   EMS203,EMS143,EMS30,EMS202,0
134 067362  076122  074206  077071  EMT204: .WORD   EMS150,EMS74,EMS202,EMS115,EMS152,EMS72,0
135 067400  073372  074415  074161  EMT205: .WORD   EMS50,EMS103,EMS72,0
136 067410  074424  074171  077071  EMT206: .WORD   EMS104,EMS73,EMS202,0
137 067420  074225  075736  074721  EMT207: .WORD   EMS75,EMS141,EMS115,EMS140,0
138 067432  074225  076122  000000  EMT210: .WORD   EMS75,EMS150,0
139 067440  073421  074161  074721  EMT211: .WORD   EMS51,EMS72,EMS115,EMS50,EMS70,0
140 067454  075764  073457  074721  EMT212: .WORD   EMS142,EMS53,EMS115,EMS143,EMS72,0
141 067470  073372  074415  073457  EMT213: .WORD   EMS50,EMS103,EMS53,0
142 067500  073447  077162  076510  EMT214: .WORD   EMS52,EMS205,EMS166,EMS206,EMS115,EMS51,EMS72,0
143 067520  073447  074754  073542  EMT215: .WORD   EMS52,EMS117,EMS56,EMS163,0
144 067532  073542  072552  073753  EMT216: .WORD   EMS56,EMS30,EMS64,0
145 067542  073314  072552  073753  EMT217: .WORD   EMS46,EMS30,EMS64,0
146 067552  072562  072552  073753  EMT220: .WORD   EMS31,EMS30,EMS64,0
147 067562  073343  072552  073753  EMT221: .WORD   EMS47,EMS30,EMS64,0
148 067572  073447  074754  074264  EMT222: .WORD   EMS52,EMS117,EMS77,0
149 067602  073447  074754  073725  EMT223: .WORD   EMS52,EMS117,EMS63,0
150 067612  000000                  EMT224: .WORD
151 067614  000000                  EMT225: .WORD
152 067616  000000                  EMT226: .WORD
153 067620  000000                  EMT227: .WORD
154 067622  000000                  EMT230: .WORD
155 067624  000000                  EMT231: .WORD
156 067626  000000                  EMT232: .WORD
157 067630  000000                  EMT233: .WORD
158 067632  000000                  EMT234: .WORD
159 067634  000000                  EMT235: .WORD
160 067636  000000                  EMT236: .WORD
161 067640  000000                  EMT237: .WORD
162 067642  000000                  EMT240: .WORD
163 067644  000000                  EMT241: .WORD
164 067646  000000                  EMT242: .WORD
165 067650  000000                  EMT243: .WORD
166 067652  000000                  EMT244: .WORD
167 067654  000000                  EMT245: .WORD
168 067656  076535  075457  077221  EMT246: .WORD   EMS167,EMS132,EMS207,0
169 067666  076535  075457  077246  EMT247: .WORD   EMS167,EMS132,EMS210,EMS125,0
170 067700  076535  077257  077246  EMT250: .WORD   EMS167,EMS211,EMS210,EMS207,EMS206,0
171 067714  077275  077320  000000  EMT251: .WORD   EMS212,EMS213,0
```

```
172 067722  076474  077275  000000  EMT252: .WORD   EMS165,EMS212,0
173 067730  076105  076141  076267  EMT253: .WORD   EMS147,EMS151,EMS156,EMS210,EMS26,EMS27,0
174 067746  077121  074340  072552  EMT254: .WORD   EMS203,EMS101,EMS30,0
175 067756  077121  076535  072552  EMT255: .WORD   EMS203,EMS167,EMS30,0
176 067766  077121  074312  072552  EMT256: .WORD   EMS203,EMS100,EMS30,0
177 067776  072136  073314  072552  EMT257: .WORD   EMS11,EMS46,EMS30,EMS102,0
178 070010  073447  074754  073542  EMT260: .WORD   EMS52,EMS117,EMS56,EMS102,0
179 070022  073447  077162  077465  EMT261: .WORD   EMS52,EMS205,EMS220,EMS206,EMS115,EMS51,EMS72,0
180 070042  074424  074171  077071  EMT262: .WORD   EMS104,EMS73,EMS202,0
181 070052  073372  073457  074366  EMT263: .WORD   EMS50,EMS53,EMS102,0
182 070062  073542  073457  074366  EMT264: .WORD   EMS56,EMS53,EMS102,EMS115,EMS150,EMS152,EMS70,0
183 070102  074146  073542  074366  EMT265: .WORD   EMS71,EMS56,EMS102,EMS115,EMS150,EMS152,EMS72,0
184 070122  075764  076001  073457  EMT266: .WORD   EMS142,EMS143,EMS53,EMS102,0
185 070134  073372  076066  074721  EMT267: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,0
186 070152  073343  073457  074366  EMT270: .WORD   EMS47,EMS53,EMS102,EMS115,EMS140,0
187 070166  073343  073457  074366  EMT271: .WORD   EMS47,EMS53,EMS102,EMS115,EMS141,EMS72,0
188 070204  074225  075736  074366  EMT272: .WORD   EMS75,EMS141,EMS102,EMS115,EMS47,EMS73,0
189 070222  073421  074206  074366  EMT273: .WORD   EMS51,EMS74,EMS102,EMS115,EMS50,EMS70,0
190 070240  073725  073457  073617  EMT274: .WORD   EMS63,EMS53,EMS57,EMS115,EMS41,EMS146,0
191 070256  074724  073457  073072  EMT275: .WORD   EMS116,EMS53,EMS41,EMS57,0
192 070270  073343  073457  073617  EMT276: .WORD   EMS47,EMS53,EMS57,EMS115,EMS140,0
193 070304  073343  073457  073617  EMT277: .WORD   EMS47,EMS53,EMS57,EMS115,EMS141,EMS72,0
194 070322  073542  073457  073617  EMT300: .WORD   EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS70,0
195 070342  074146  073542  073457  EMT301: .WORD   EMS71,EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS72,0
196 070364  076474  074312  074415  EMT302: .WORD   EMS165,EMS100,EMS103,EMS57,0
197 070376  076474  074340  074415  EMT303: .WORD   EMS165,EMS101,EMS103,EMS57,0
198 070410  076474  074264  074415  EMT304: .WORD   EMS165,EMS77,EMS103,EMS57,0
199 070422  073314  072552  073753  EMT305: .WORD   EMS46,EMS30,EMS64,EMS57,0
200 070434  073372  073457  073617  EMT306: .WORD   EMS50,EMS53,EMS57,0
201 070444  073372  076066  074721  EMT307: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,EMS57,0
202 070464  075764  076001  073457  EMT310: .WORD   EMS142,EMS143,EMS53,EMS57,0
203 070476  074424  074415  073457  EMT311: .WORD   EMS104,EMS103,EMS53,EMS57,0
204 070510  074453  074415  073457  EMT312: .WORD   EMS105,EMS103,EMS53,EMS57,0
205 070522  076027  076037  074415  EMT313: .WORD   EMS144,EMS145,EMS103,EMS57,EMS115,EMS143,EMS70,0
206 070542  073120  074415  073457  EMT314: .WORD   EMS42,EMS103,EMS53,EMS57,0
207 070554  072562  074415  073457  EMT315: .WORD   EMS31,EMS103,EMS53,EMS57,0
208 070566  074146  072562  074415  EMT316: .WORD   EMS71,EMS31,EMS103,EMS57,0
209 070600  073147  074415  073617  EMT317: .WORD   EMS43,EMS103,EMS57,0
210 070610  073251  074415  073617  EMT320: .WORD   EMS45,EMS103,EMS57,0
211 070620  073176  074415  073617  EMT321: .WORD   EMS44,EMS103,EMS57,0
212 070630  074502  072344  000000  EMT322: .WORD   EMS106,EMS20,0
213 070636  072766  074415  073617  EMT323: .WORD   EMS36,EMS103,EMS57,0
214 070646  076664  072766  074415  EMT324: .WORD   EMS173,EMS36,EMS103,EMS57,0
215 070660  076644  072766  074415  EMT325: .WORD   EMS172,EMS36,EMS103,EMS57,0
216 070672  072210  076701  072232  EMT326: .WORD   EMS13,EMS174,EMS15,EMS35,EMS53,EMS175,0
217 070710  076105  076141  074206  EMT327: .WORD   EMS147,EMS151,EMS74,EMS175,0
218 070722  074037  073457  076707  EMT330: .WORD   EMS66,EMS53,EMS175,0
219 070732  072640  074415  073457  EMT331: .WORD   EMS33,EMS103,EMS53,EMS175,0
220 070744  073043  074415  073457  EMT332: .WORD   EMS40,EMS103,EMS53,EMS57,0
221 070756  073421  074206  073617  EMT333: .WORD   EMS51,EMS74,EMS57,EMS115,EMS50,EMS70,0
222 070774  074225  075736  073617  EMT334: .WORD   EMS75,EMS141,EMS57,EMS115,EMS47,EMS73,0
223 071012  074225  076122  076165  EMT335: .WORD   EMS75,EMS150,EMS152,EMS57,EMS115,EMS56,EMS73,0
224 071032  073645  073660  073707  EMT336: .WORD   EMS60,EMS61,EMS62,0
225 071042  074724  074754  072666  EMT337: .WORD   EMS116,EMS117,EMS34,0
226 071052  072666  073457  073471  EMT340: .WORD   EMS34,EMS53,EMS54,EMS111,0
227 071064  073513  072666  000000  EMT341: .WORD   EMS55,EMS34,0
228 071072  073447  074754  073542  EMT342: .WORD   EMS52,EMS117,EMS56,EMS57,0
```

```
    229 071104  073447  074754  073251  EMT343: .WORD   EMS52,EMS117,EMS45,EMS57,0
    230 071116  073447  074754  073176  EMT344: .WORD   EMS52,EMS117,EMS44,EMS57,0
    231 071130  073447  074754  077505  EMT345: .WORD   EMS52,EMS117,EMS221,0
    232 071140  074502  072344  074721  EMT346: .WORD   EMS106,EMS20,EMS115,EMS223,EMS72,0
    233 071154  073447  077162  077555  EMT347: .WORD   EMS52,EMS205,EMS222,EMS206,0
    234 071166  074225  076122  074366  EMT350: .WORD   EMS75,EMS150,EMS102,EMS115,EMS56,EMS73,0
    235 071204  076535  074171  074366  EMT351: .WORD   EMS167,EMS73,EMS102,0
    236 071214  077361  000000          EMT352: .WORD   EMS215,0
    237 071220  077432  077121  077402  EMT353: .WORD   EMS217,EMS203,EMS216,0
    238 071230  077505  072344  000000  EMT354: .WORD   EMS221,EMS20,0
    239
```

```
 1 071236  077735  100541  100616  EHT1:   .WORD   EH1,STSH1,STSH2,STSH4,0
 2 071250  100541  100616  100743  EHT2:   .WORD   STSH1,STSH2,STSH4,0
 3
 4 071260  077754  000000          EHT110: .WORD   EH110,0
 5 071264  077763  000000          EHT111: .WORD   EH111,0
 6
 7 071270  100002  000000          EHT114: .WORD   EH114,0
 8 071274  100031  100541  100616  EHT223: .WORD   EH223,STSH1,STSH2,STSH4,0
 9 071306  100057  100541  100616  EHT256: .WORD   EH256,STSH1,STSH2,STSH4,0
10
11 071320  100133  100541  100616  EHT336: .WORD   EH336,STSH1,STSH2,STSH4,0
12 071332  100172  100541  100616  EHT337: .WORD   EH337,STSH1,STSH2,STSH4,0
13 071344  100327  100541  100616  EHT344: .WORD   EH344,STSH1,STSH2,STSH4,0
14
15 071356  100465  000000          EHT353: .WORD   EH353,0
16
```

```
 1 071362  101002  101076  101114  EDT1:    .WORD   ED1,STSD1,STSD2,STSD4
 2 071372  101076  101114  101146  EDT2:    .WORD   STSD1,STSD2,STSD4
 3
 4 071400  101010                  EDT110:  .WORD   ED110
 5 071402  101014                  EDT111:  .WORD   ED111
 6
 7 071404  101022                  EDT114:  .WORD   ED114
 8 071406  101032  101076  101114  EDT223:  .WORD   ED223,STSD1,STSD2,STSD4
 9
10 071416  101042  101076  101114  EDT336:  .WORD   ED336,STSD1,STSD2,STSD4
11 071426  101054  101076  101114  EDT337:  .WORD   ED337,STSD1,STSD2,STSD4
12 071436  101054  101076  101114  EDT344:  .WORD   ED337,STSD1,STSD2,STSD4,0
13
14 071450  101066                  EDT353:  .WORD   ED353
15
```

ERROR MESSAGE TABLE

SEQ 0258

```
 1 071452  101161  101177  101177  EFT1:    .WORD   EF111,STSF,STSF,STSF
 2 071462  101177  101177  101177  EFT2:    .WORD   STSF,STSF,STSF
 3
 4 071470  101160                  EFT110:  .WORD   EF110
 5 071472  101161                  EFT111:  .WORD   EF111
 6
 7 071474  101163                  EFT114:  .WORD   EF114
 8 071476  101163  101177  101177  EFT223:  .WORD   EF114,STSF,STSF,STSF
 9
10 071506  101166  101177  101177  EFT336:  .WORD   EF336,STSF,STSF,STSF
11 071516  101166  101177  101177  EFT337:  .WORD   EF336,STSF,STSF,STSF
12 071526  101166  101177  101177  EFT344:  .WORD   EF336,STSF,STSF,STSF
13
14 071536  101163                  EFT353:  .WORD   EF114
15
```

```
                                .SBTTL  ERROR MESSAGE STRINGS
                                ;
071540  127  122  117  EMS1:   .ASCIZ  @WRONG UNIT SELECTED (RMCS2, BITS 0-2) @
071607  104  105  126  EMS2:   .ASCIZ  @DEVICE WENT @
071624  125  116  101  EMS3:   .ASCIZ  @UNAVAILABLE "DVA" (RMCS1, BIT 11) @
071667  116  117  116  EMS4:   .ASCIZ  @NONEXISTENT "NED" (RMCS2, BIT 12) @
071732  103  117  115  EMS5:   .ASCIZ  @COMMAND NOT COMPLETED, @
071762  103  117  116  EMS6:   .ASCIZ  @CONTROLLER NOT READY (RMCS1, BIT 7) @
072027  104  122  111  EMS7:   .ASCIZ  @DRIVE NOT READY "DRY" (RMDS, BIT 7) @
072074  107  117  040  EMS10:  .ASCIZ  @GO NOT RESET "GO" (RMCS1, BIT 0) @
072136  111  116  126  EMS11:  .ASCIZ  @INVALID @
072147  106  125  116  EMS12:  .ASCIZ  @FUNCTION CODE (RMCS1, BITS 1-5) @
072210  115  101  123  EMS13:  .ASCIZ  @MASSBUS @
072221  103  117  116  EMS14:  .ASCIZ  @CONTROL @
072232  102  125  123  EMS15:  .ASCIZ  @BUS PARITY ERROR @
072254  042  115  103  EMS16:  .ASCIZ  @"MCPE" (RMCS1, BIT 13) @
072304  124  122  101  EMS17:  .ASCIZ  @TRANSFER ERROR (RMCS1, BIT 14) @
072344  123  110  117  EMS20:  .ASCIZ  @SHOULD NOT BE SET @
072367  127  117  122  EMS21:  .ASCIZ  @WORD COUNT (RMWC) @
072412  102  125  123  EMS22:  .ASCIZ  @BUS (RMBA) @
072426  042  114  102  EMS23:  .ASCIZ  @"LBT" (RMDS, BIT 10) @
072454  042  101  117  EMS24:  .ASCIZ  @"AOE" (RMER1, BIT 09) @
072503  104  111  123  EMS25:  .ASCIZ  @DISK (RMDA) @
072520  103  131  114  EMS26:  .ASCIZ  @CYLINDER (RMDC) @
072541  101  104  104  EMS27:  .ASCIZ  @ADDRESS @
072552  123  124  101  EMS30:  .ASCIZ  @STATUS @
072562  042  127  114  EMS31:  .ASCIZ  @"WLE" (RMER1, BIT 11) @
072611  042  125  120  EMS32:  .ASCIZ  @"UPE" (RMCS2, BIT 13) @
072640  042  127  103  EMS33:  .ASCIZ  @"WCF" (RMER1, BIT 5) @
072666  127  122  111  EMS34:  .ASCIZ  @WRITE CHECK ERROR-"WCE" (RMCS2, BIT 14) @
072737  042  115  104  EMS35:  .ASCIZ  @"MDPE" (RMCS2, BIT 8) @
072766  042  104  103  EMS36:  .ASCIZ  @"DCK" (RMER1, BIT 15) @
073015  042  105  103  EMS37:  .ASCIZ  @"ECH" (RMER1, BIT 6) @
073043  042  104  114  EMS40:  .ASCIZ  @"DLT" (RMCS2, BIT 15) @
073072  042  115  130  EMS41:  .ASCIZ  @"MXF" (RMCS2, BIT 9) @
073120  042  104  124  EMS42:  .ASCIZ  @"DTE" (RMER1, BIT 12) @
073147  042  110  103  EMS43:  .ASCIZ  @"HCRC" (RMER1, BIT 8) @
073176  110  105  101  EMS44:  .ASCIZ  @HEADER COMPARE ERROR "HCE" (RMER1, BIT 7) @
073251  106  117  122  EMS45:  .ASCIZ  @FORMAT ERROR "FER" (RMER1, BIT 4) @
073314  042  111  101  EMS46:  .ASCIZ  @"IAE" (RMER1, BIT 10) @
073343  042  117  120  EMS47:  .ASCIZ  @"OPI" (RMER1, BIT 13) @
073372  042  123  113  EMS50:  .ASCIZ  @"SKI" (RMER2, BIT 14) @
073421  042  120  111  EMS51:  .ASCIZ  @"PIP" (RMDS, BIT 13) @
073447  124  110  105  EMS52:  .ASCIZ  @THE RM @
073457  104  105  124  EMS53:  .ASCIZ  @DETECTED @
073471  101  124  040  EMS54:  .ASCIZ  @AT AN UNEXPECTED @
073513  111  116  103  EMS55:  .ASCIZ  @INCORRECT DATA DURING @
073542  111  116  126  EMS56:  .ASCIZ  @INVALID COMMAND ERROR "IVC" (RMER2, BIT 12) @
073617  104  125  122  EMS57:  .ASCIZ  @DURING DATA TRANSFER @
073645  104  101  124  EMS60:  .ASCIZ  @DATA READ @
073660  104  117  105  EMS61:  .ASCIZ  @DOES NOT COMPARE WITH @
073707  104  101  124  EMS62:  .ASCIZ  @DATA WRITTEN @
073725  042  120  101  EMS63:  .ASCIZ  @"PAR" (RMER1, BIT 3) @
073753  111  123  040  EMS64:  .ASCIZ  @IS INCORRECT @
073771  103  117  115  EMS65:  .ASCIZ  @COMPOSITE ERROR "ERR" (RMDS, BIT 14) @
074037  104  101  124  EMS66:  .ASCIZ  @DATA PARITY ERROR "DPE" (RMER2, BIT 3) @
074107  104  125  122  EMS67:  .ASCIZ  @DURING SEEK COMMAND @
```

```
 58 074134  111  123  040  EMS70:  .ASCIZ  @IS RESET @
 59 074146  105  122  122  EMS71:  .ASCIZ  @ERRONEOUS @
 60 074161  111  123  040  EMS72:  .ASCIZ  @IS SET @
 61 074171  104  111  104  EMS73:  .ASCIZ  @DID NOT SET @
 62 074206  104  111  104  EMS74:  .ASCIZ  @DID NOT RESET @
 63 074225  114  117  123  EMS75:  .ASCIZ  @LOST @
 64 074233  104  125  122  EMS76:  .ASCIZ  @DURING PACK ACK COMMAND @
 65 074264  042  122  115  EMS77:  .ASCIZ  @"RMR" (RMER1, BIT 2) @
 66 074312  042  111  114  EMS100: .ASCIZ  @"ILR" (RMER1, BIT 1) @
 67 074340  042  111  114  EMS101: .ASCIZ  @"ILF" (RMER1, BIT 0) @
 68 074366  104  125  122  EMS102: .ASCIZ  @DURING SEARCH COMMAND @
 69 074415  105  122  122  EMS103: .ASCIZ  @ERROR @
 70 074424  042  114  102  EMS104: .ASCIZ  @"LBC" (RMER2, BIT 10) @
 71 074453  042  114  123  EMS105: .ASCIZ  @"LSC" (RMER2, BIT 11) @
 72 074502  110  105  101  EMS106: .ASCIZ  @HEADER ERRORS @
 73 074521  102  125  123  EMS110: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
 74 074550  101  104  104  EMS111: .ASCIZ  @ADDRESS @
 75 074561  127  110  105  EMS112: .ASCII  @WHEN READING/WRITING RH REGISTERS @
 76 074623  101  124  040           .ASCIZ  @AT THE FOLLOWING @
 77 074645  124  110  105  EMS113: .ASCIZ  @THE SELECTED DEVICE IS @
 78 074675  116  117  116  EMS114: .ASCIZ  @NONEXISTENT DEVICE @
 79 074721  015  012  000  EMS115: .ASCIZ  <CR><LF>
 80 074724  124  110  105  EMS116: .ASCIZ  @THE MASSBUS CONTROLLER @
 81 074754  106  101  111  EMS117: .ASCIZ  @FAILED TO DETECT @
 82 074776  116  117  124  EMS120: .ASCIZ  @NOT AN RM05/3/2 @
 83 075017  103  117  116  EMS121: .ASCIZ  @CONTROL STATUS REGISTER 1, RMCS1, @
 84 075062  102  125  123  EMS122: .ASCIZ  @BUS ADDRESS REGISTER, RMBA, @
 85 075117  103  117  116  EMS123: .ASCIZ  @CONTROL STATUS REGISTER 2, RMCS2, @
 86 075162  105  122  122  EMS124: .ASCIZ  @ERROR REGISTER 1, RMER1, @
 87 075214  101  124  124  EMS125: .ASCIZ  @ATTENTION SUMMARY REGISTER, RMAS, @
 88 075257  115  101  111  EMS126: .ASCIZ  @MAINTENANCE REGISTER #1, RMMR #1, @
 89 075322  105  103  103  EMS127: .ASCIZ  @ECC POSITION REGISTER, RMEC1, @
 90 075361  105  103  103  EMS130: .ASCIZ  @ECC PATTERN REGISTER, RMEC2, @
 91 075417  115  101  111  EMS131: .ASCIZ  @MAINTENANCE REGISTER 2, RMMR2, @
 92 075457  116  117  124  EMS132: .ASCIZ  @NOT INITIALIZED BY @
 93 075503  125  116  111  EMS133: .ASCIZ  @UNIBUS INITIALIZE @
 94 075526  103  117  116  EMS134: .ASCIZ  @CONTROLLER CLEAR, I.E.  BIT 5 OF @
 95 075570  122  110  057  EMS135: .ASCIZ  @RH/RM ERROR CLEAR (RMCS1, BIT 14) @
 96 075633  104  122  111  EMS136: .ASCIZ  @DRIVE CLEAR COMMAND @
 97 075660  104  122  111  EMS137: .ASCIZ  @DRIVE STATUS REGISTER, RMDS @
 98 075715  115  105  104  EMS140: .ASCIZ  @MEDIUM OFF LINE @
 99 075736  042  115  117  EMS141: .ASCIZ  @"MOL" (RMDS, BIT 12) @
100 075764  104  122  111  EMS142: .ASCIZ  @DRIVE FAULT @
101 076001  042  104  126  EMS143: .ASCIZ  @"DVC" (RMER2, BIT 7) @
102 076027  125  116  123  EMS144: .ASCIZ  @UNSAFE @
103 076037  042  125  116  EMS145: .ASCIZ  @"UNS" (RMER1, BIT 14) @
104 076066  123  110  117  EMS146: .ASCIZ  @SHOULD BE SET @
105 076105  117  106  106  EMS147: .ASCIZ  @OFFSET MODE @
106 076122  040  126  117  EMS150: .ASCIZ   @ VOLUME VALID @
107 076141  042  117  115  EMS151: .ASCIZ  @"OM" (RMDS, BIT 0) @
108 076165  042  126  126  EMS152: .ASCIZ  @"VV" (RMDS, BIT 6) @
109 076211  120  101  103  EMS153: .ASCIZ  @PACK ACK COMMAND @
110 076233  116  117  124  EMS154: .ASCIZ  @NOT SET BY @
111 076247  117  106  106  EMS155: .ASCIZ  @OFFSET COMMAND @
112 076267  116  117  124  EMS156: .ASCIZ  @NOT RESET BY @
113 076305  122  124  103  EMS157: .ASCIZ  @RTC COMMAND @
114 076322  122  111  120  EMS160: .ASCIZ  @RIP COMMAND @
```

```
115 076337      117     106     106  EMS161: .ASCIZ  @OFFSET REGISTER (RMOF) @
116 076367                           EMS162:         ;<UNUSED>
117 076367      104     125     122  EMS163: .ASCIZ  @DURING RECALIBRATE @
118 076413      111     123     040  EMS164: .ASCII  @IS INTERMITTENT OR DRIVE DIDNT DROP ON @
119 076462      103     131     114          .ASCIZ  @CYLINDER @
120 076474      125     116     105  EMS165: .ASCIZ  @UNEXPECTED @
121 076510      122     105     103  EMS166: .ASCIZ  @RECALIBRATE COMMAND @
122 076535      042     101     124  EMS167: .ASCIZ  @"ATA" (RMDS, BIT15) @
123 076562      127     110     105  EMS170: .ASCIZ  @WHEN READING REGISTER @
124 076611      105     122     122  EMS171: .ASCIZ  @ERROR REGISTER #2, RMER2, @
125 076644      116     117     116  EMS172: .ASCIZ  @NONRECOVERABLE @
126 076664      122     105     103  EMS173: .ASCIZ  @RECOVERABLE @
127 076701      104     101     124  EMS174: .ASCIZ  @DATA @
128 076707      104     125     122  EMS175: .ASCIZ  @DURING WRITE COMMAND @
129 076735      042     117     120  EMS176: .ASCIZ  @"OPE" (RMER2, BIT 13) @
130 076764      111     116     040  EMS177: .ASCIZ  @IN WRITE PROTECT @
131 077006      103     101     116  EMS200: .ASCIZ  @CAN NOT SET @
132 077023      104     111     101  EMS201: .ASCIZ  @DIAGNOSTIC MODE "DMD" (RMMR1, BIT 0) @
133 077071      104     125     122  EMS202: .ASCIZ  @DURING DIAGNOSTIC MODE @
134 077121      111     116     103  EMS203: .ASCIZ  @INCORRECT @
135 077134      042     127     122  EMS204: .ASCIZ  @"WRL" (RMDS, BIT 11) @
136 077162      105     130     105  EMS205: .ASCIZ  @EXECUTED @
137 077174      127     111     124  EMS206: .ASCIZ  @WITH COMP ERROR SET @
138 077221      042     107     117  EMS207: .ASCIZ  @"GO" (RMCS1, BIT 0) @
139 077246      127     122     111  EMS210: .ASCIZ  @WRITING @
140 077257      127     101     123  EMS211: .ASCIZ  @WAS RESET BY @
141 077275      120     122     117  EMS212: .ASCIZ  @PROGRAM INTERRUPT @
142 077320      127     101     123  EMS213: .ASCIZ  @WAS NOT GENERATED @
143 077343      123     105     105  EMS214: .ASCIZ  @SEEK COMMAND @
144 077361      120     122     117  EMS215: .ASCIZ  @PROGRAM TIMEOUT @
145 077402      104     125     122  EMS216: .ASCIZ  @DURING LOOK AHEAD TEST @
146 077432      114     117     117  EMS217: .ASCIZ  @LOOK AHEAD REGISTER,RMLA, @
147 077465      123     105     101  EMS220: .ASCIZ  @SEARCH COMMAND @
148 077505      102     101     104  EMS221: .ASCIZ  @BAD SECTOR ERROR "BSE" (RMER2, BIT 15) @
149 077555      101     040     104  EMS222: .ASCIZ  @A DATA TRANSFER COMMAND @
150 077606      110     105     101  EMS223: .ASCIZ  @HEADER COMPARE INHIBIT "HCI" (RMOF, BIT 10) @
151 077663      116     117     116  EMS224: .ASCIZ  @NONEXISTENT MEMORY "NEM" (RMCS2, BIT 11) @
152
```

```
      1 077735      105     130     120  EH1:    .ASCIZ  @EXPCTD  RECEVD@
      2 077754      102     125     123  EH110:  .ASCIZ  @BUSADR@
      3 077763      040     122     115  EH111:  .ASCIZ  @ RMCS2   RMCS1@
      4
      5 100002      122     105     103  EH114:  .ASCIZ  @RECEVD  SNGPRT  DULPRT@
      6 100031      105     130     120  EH223:  .ASCIZ  @EXPCTD  RECEVD   DATA@
      7 100057      105     130     120  EH256:  .ASCII  @EXPCTD  RECEVD  RGSTR@<CRLF>
      8 100105      123     124     101          .ASCIZ  @STATUS  STATUS  INDEX@
      9
     10 100133      107     104     101  EH336:  .ASCIZ  @GDADRS  GDDATA  BDADRS  BDDATA@
     11 100172      122     115     103  EH337:  .ASCII  @RMCS2   STATUS  FAILING   DATA@<CRLF>
     12 100231      137     137     137          .ASCII  @                              @<CRLF>
     13 100270      105     130     120          .ASCIZ  @EXPCTD--RECEVD  --BIT---ADRESS@
     14
     15 100327      122     115     105  EH344:  .ASCII  @RMER1   STATUS  HEADER  FAILING@<CRLF>
     16 100367      137     137     137          .ASCII  @                WORD     BIT@<CRLF>
     17 100425      105     130     120          .ASCIZ  @EXPCTD--RECEVD  NUMBER  POSITON@
     18 100465      105     130     120  EH353:  .ASCII  @EXPCTD  RECEVD@<CRLF>
     19 100504      074     103     122          .ASCIZ  @<CRLF> RMLA    RMLA    RMOF @
     20
     21 100541      040     122     115  STSH1:  .ASCII  @ RMCS1  RMCS2   RMDS    RMER1   RMER2@
     22 100606      040     040     040          .ASCIZ  @   RMAS@
     23 100616      040     122     115  STSH2:  .ASCII  @ RMWC    RMBA    RMDA    RMOF    RMDC@
     24 100663      040     040     040          .ASCIZ  @    RMEC1   RMEC2@
     25 100705      040     122     115  STSH3:  .ASCIZ  @ RMDA    RMDC    RMOF    RMLA@
     26 100743      040     122     115  STSH4:  .ASCIZ  @ RMMR1   RMMR2   RMDT    RMSN@
     27                                  .EVEN
     28
```

```
      1 101002  001140  001142  000000  ED1:    .WORD   $GDDAT,$BDDAT,0
      2 101010  001276  000000          ED110:  .WORD   $BASE,0
      3 101014  001174  001176  000000  ED111:  .WORD   $TMP0,$TMP1,0
      4
      5 101022  001362  001176  001200  ED114:  .WORD   RMDTI,$TMP1,$TMP2,0
      6 101032  001140  001142  001174  ED223:  .WORD   $GDDAT,$BDDAT,$TMP0,0
      7
      8 101042  001134  001140  001136  ED336:  .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
      9
     10 101054  001140  001142  001174  ED337:  .WORD   $GDDAT,$BDDAT,$TMP0,$TMP1,0
     11 101066  001140  001142  001442  ED353:  .WORD   $GDDAT,$BDDAT,RMOF0,0
     12
     13 101076  001334  001344  001346  STSD1:  .WORD   RMCS1I,RMCS2I,RMDSI,RMER1I,RMER2I,RMASI,0
     14 101114  001336  001340  001342  STSD2:  .WORD   RMWCI,RMBAI,RMDAI,RMOFI,RMDCI,RMEC1I
     15 101130  001402  000000          .WORD   RMEC2I,0
     16 101134  001342  001370  001366  STSD3:  .WORD   RMDAI,RMDCI,RMOFI,RMLAI,0
     17 101146  001360  001374  001362  STSD4:  .WORD   RMMR1I,RMMR2I,RMDTI,RMSNI,0
     18
```

```
  101160   000                 EF110:  .BYTE   0
  101161   000     000         EF111:  .BYTE   0,0
  101163   000     000    000  EF114:  .BYTE   0,0,0
  101166   000     000    000  EF336:  .BYTE   0,0,0,0
  101172   000     000    000  EF337:  .BYTE   0,0,0,0,0

  101177   000     000    000  STSF:   .BYTE   0,0,0,0,0,0,0
                               .EVEN

```

```
                                        ;STORAGE FOR GENERAL DATA TRANSFERRS
101206                                  BUFFER:
101206                                  BUFONE: .BLKW   258.
102212                                  BUFTWO: .BLKW   258.

                                        ;STORAGE FOR MANUFACTURES 16 BIT MODE BAD SECTOR FILE
103216  000000  000000                  MFGFIL: .WORD   0,0              ;2 HEADER WORDS
                                                .BLKW   256.            ;256. WORDS OF DATA
104222  177777                                  .WORD   -1              ;TERMINATOR IF FILE IS FULL

                                        ;STORAGE FOR USERS 16 BIT MODE BAD SECTOR FILE
104224  000000  000000                  USRFIL: .WORD   0,0              ;2 HEADER WORDS
                                                .BLKW   256.            ;256. WORDS OF DATA
105230  177777                                  .WORD   -1              ;TERMINATOR IF FILE IS FULL

        101206                          .=BUFFER

101206                                  HELP:
101206      200                         .ASCII  <CRLF>
101207      200                         .ASCII  <CRLF>
101210      114     111     123         .ASCII  @LIST OF TESTS@<CRLF>
101226      055     055     055         .ASCII  @------------@<CRLF>
101244      124     061     011         .ASCII  @T1     CONTROLLER ACCESS TEST@<CRLF>
101276      124     062     011         .ASCII  @T2     DEVICE AVAILABLE TEST@<CRLF>
101327      124     063     011         .ASCII  @T3     DRIVE TYPE TEST@<CRLF>
101352      124     064     011         .ASCII  @T4     FORMAT ZEROS@<CRLF>
101372      124     065     011         .ASCII  @T5     ZERO FILL TEST@<CRLF>
101414      124     066     011         .ASCII  @T6     FORMAT CHECK ZEROS@<CRLF>
101442      124     067     011         .ASCII  @T7     FORMAT CHECK ZEROS W/ WCE ERROR@<CRLF>
101505      124     061     060         .ASCII  @T10    FORMAT ONES@<CRLF>
101525      124     061     061         .ASCII  @T11    FORMAT CHECK ONES@<CRLF>
101553      124     061     062         .ASCII  @T12    FORMAT CHECK ONES W/ WCE ERRORS@<CRLF>
101617      124     061     063         .ASCII  @T13    FORMAT MULTIPLE SECTORS@<CRLF>
101653      124     061     064         .ASCII  @T14    FORMAT W/ HEAD SWITCHING@<CRLF>
101710      124     061     065         .ASCII  @T15    FORMAT W/ MID TRANSFER SEEK@<CRLF>
101750      124     061     066         .ASCII  @T16    FORMAT W/ IMPLIED SEEK@<CRLF>
102003      124     061     067         .ASCII  @T17    FORMAT EACH SECTOR ADDRESS@<CRLF>
102042      124     062     060         .ASCII  @T20    FORMAT EACH TRACK ADDRESS@<CRLF>
102100      124     062     061         .ASCII  @T21    FORMAT PRIME CYLINDERS@<CRLF>
102133      124     062     062         .ASCII  @T22    READ HEADER & DATA IN LAST SECTOR@<CRLF>
102201      124     062     063         .ASCII  @T23    READ HEADER & DATA W/ AOE ERROR@<CRLF>
102245      124     062     064         .ASCII  @T24    READ INVALID SECTOR ADDRESS@<CRLF>
102305      124     062     065         .ASCII  @T25    READ INVALID TRACK ADDRESS@<CRLF>
102344      124     062     066         .ASCII  @T26    READ INVALID CYLINDER ADDRESS@<CRLF>
102406      124     062     067         .ASCII  @T27    FORMAT AT OFFSET@<CRLF>
102433      124     063     060         .ASCII  @T30    IVC FORMAT TEST@<CRLF>
102457      124     063     061         .ASCII  @T31    FORMAT ERROR TEST@<CRLF>
102505      124     063     062         .ASCII  @T32    FORMAT HCE, FIRST HEADER WORD@<CRLF>
102547      124     063     063         .ASCII  @T33    FORMAT HCE, SECOND HEADER WORD@<CRLF>
102612      200                         .ASCII  <CRLF>
102613      117     120     105         .ASCII  @OPERATIONAL SWITCH SETTINGS@<CRLF>
102647      055     055     055         .ASCII  @---------------------------@<CRLF>
102703      123     127     111         .ASCII  @SWITCH                  USE@<CRLF>
102720      055     055     055         .ASCII  @------         -------------------@<CRLF>
102755      040     040     061         .ASCII  @  15           HALT ON ERROR@<CRLF>
103001      040     040     061         .ASCII  @  14           LOOP ON TEST@<CRLF>
103024      040     040     061         .ASCII  @  13           INHIBIT ERROR TYPEOUTS@<CRLF>
```

```
                                                                    1  5
     58 103061      040      040      061   .ASCII  @  12          @<CRLF>
     59 103070      040      040      061   .ASCII  @  11          INHIBIT ITERATIONS@<CRLF>
     60 103121      040      040      061   .ASCII  @  10          BELL ON ERROR@<CRLF>
     61 103145      040      040      040   .ASCII  @   9          LOOP ON ERROR@<CRLF>
     62 103171      040      040      040   .ASCII  @   8          LOOP ON TEST IN SWR<7:0>@<CRLF>
     63 103230      040      040      040   .ASCII  @   7          TN128@<CRLF>
     64 103244      040      040      040   .ASCII  @   6          TN64@<CRLF>
     65 103257      040      040      040   .ASCII  @   5          TN32@<CRLF>
     66 103272      040      040      040   .ASCII  @   4          TN16@<CRLF>
     67 103305      040      040      040   .ASCII  @   3          TN8@<CRLF>
     68 103317      040      040      040   .ASCII  @   2          TN4@<CRLF>
     69 103331      040      040      040   .ASCII  @   1          TN2@<CRLF>
     70 103343      040      040      040   .ASCIZ  @   0          TN1@<CRLF>
     71
     72          000200                    .END    200
```

SYMBOL TABLE

```
ABASE = 176700        ATNMSK= 000377        CHRCNT  033313        EDT1    071362        EMS100  074312
ACDW1 = 000000        ATNTBL  064506        CKSWR = 104410        EDT110  071400        EMS101  074340
ACDW2 = 000000        AUNIT = 000000        CLKADR  001516        EDT111  071402        EMS102  074366
ACKSTS  047462        AUSWR = 000000        CLKVCT  001520        EDT114  071404        EMS103  074415
ACPUOP= 000000        AVECT1= 120254        CLR   = 000040        EDT2    071372        EMS104  074424
ADDW0 = 000000        AVECT2= 000000        CLRSTS  046602        EDT223  071406        EMS105  074453
ADDW1 = 000000        A16   = 000400        CMNSTA  007612        EDT336  071416        EMS106  074502
ADDW10= 000000        A17   = 001000        CMPBUF  036462        EDT337  071426        EMS11   072136
ADDW11= 000000        BACK  = 000000        CMPERR  044600        EDT344  071436        EMS110  074521
ADDW12= 000000        BADSCT  034276        CNSL00  063644        EDT353  071450        EMS111  074550
ADDW13= 000000        BADTMO  005340        CNSL01  063677        ED1     101002        EMS112  074561
ADDW14= 000000        BA1   = 000010        CNSL02  063715        ED110   101010        EMS113  074645
ADDW15= 000000        BB00  = 000001        CNSL03  063757        ED111   101014        EMS114  074675
ADDW2 = 000000        BB01  = 000002        CNSL04  063777        ED114   101022        EMS115  074721
ADDW3 = 000000        BB02  = 000004        CNSL05  064033        ED223   101032        EMS116  074724
ADDW4 = 000000        BB03  = 000010        CNSL06  064045        ED336   101042        EMS117  074754
ADDW5 = 000000        BB04  = 000020        CNSL07  064076        ED337   101054        EMS12   072147
ADDW6 = 000000        BB05  = 000040        CNSL08  064241        ED353   101066        EMS120  074776
ADDW7 = 000000        BB06  = 000100        CNSL09  064242        EECC  = 000020        EMS121  075017
ADDW8 = 000000        BB07  = 000200        CNTCLR  046464        EFT1    071452        EMS122  075062
ADDW9 = 000000        BB08  = 000400        COMMA   063231        EFT110  071470        EMS123  075117
ADEVCT= 000000        BB09  = 001000        CONT  = 000100        EFT111  071472        EMS124  075162
ADEVM = 000000        BIT0  = 000001        CR    = 000015        EFT114  071474        EMS125  075214
ADR   = 000001        BIT00 = 000001        CRLF  = 000200        EFT2    071462        EMS126  075257
AENV  = 000000        BIT01 = 000002        CTLFG   001326        EFT223  071476        EMS127  075322
AENVM = 000000        BIT02 = 000004        CYLMSK= 001777        EFT336  071506        EMS13   072210
AFATAL= 000000        BIT03 = 000010        DBCK  = 100000        EFT337  071516        EMS130  075361
ALL     063220        BIT04 = 000020        DBEN  = 040000        EFT344  071526        EMS131  075417
AMADR1= 000000        BIT05 = 000040        DBL   = 002000        EFT353  071536        EMS132  075457
AMADR2= 000000        BIT06 = 000100        DCK   = 100000        EF110   101160        EMS133  075503
AMADR3= 000000        BIT07 = 000200        DDISP = 177570        EF111   101161        EMS134  075526
AMADR4= 000000        BIT08 = 000400        DEBL  = 020000        EF114   101163        EMS135  075570
AMAMS1= 000000        BIT09 = 001000        DEVSEL  045012        EF336   101166        EMS136  075633
AMAMS2= 000000        BIT1  = 000002        DISPLA  001156        EF337   101172        EMS137  075660
AMAMS3= 000000        BIT10 = 002000        DISPRE  000174        EHT1    071236        EMS14   072221
AMAMS4= 000000        BIT11 = 004000        DLT   = 100000        EHT110  071260        EMS140  075715
AMSGAD= 000000        BIT12 = 010000        DMD   = 000001        EHT111  071264        EMS141  075736
AMSGLG= 000000        BIT13 = 020000        DPE   = 000010        EHT114  071270        EMS142  075764
AMSGTY= 000000        BIT14 = 040000        DPEHI = 040000        EHT2    071250        EMS143  076001
AMTYP1= 000000        BIT15 = 100000        DPELO = 020000        EHT223  071274        EMS144  076027
AMTYP2= 000000        BIT2  = 000004        DPR   = 000400        EHT256  071306        EMS145  076037
AMTYP3= 000000        BIT3  = 000010        DRQ   = 004000        EHT336  071320        EMS146  076066
AMTYP4= 000000        BIT4  = 000020        DRVCLR= 000010        EHT337  071332        EMS147  076105
AOE   = 001000        BIT5  = 000040        DRVSTS  052020        EHT344  071344        EMS15   072232
APASS = 000000        BIT6  = 000100        DRY   = 000200        EHT353  071356        EMS150  076122
APE   = 100000        BIT7  = 000200        DSWR  = 177570        EH1     077735        EMS151  076141
APRIOR= 000000        BIT8  = 000400        DTASTS  052622        EH110   077754        EMS152  076165
APTCSU= 000040        BIT9  = 001000        DTE   = 010000        EH111   077763        EMS153  076211
APTENV= 000001        BOTADR  033310        DTO   = 010000        EH114   100002        EMS154  076233
APTSIZ= 000200        BOTFLG  033312        DULPRT= 024024        EH223   100031        EMS155  076247
APTSPO= 000100        BPTVEC= 000014        DVA   = 004000        EH256   100057        EMS156  076267
ARGS  = 000004        BSE   = 100000        DVC   = 000200        EH336   100133        EMS157  076305
ASNDA   001514        BUFFER  101206        EARLY   065032        EH337   100172        EMS16   072254
ASNDC   001512        BUFONE  101206        EBL   = 020000        EH344   100327        EMS160  076322
ASWREG= 000000        BUFTWO  102212        ECH   = 000100        EH353   100465        EMS161  076337
ATA   = 100000        CC    = 004000        ECI   = 004000        EMS1    071540        EMS162  076367
ATESTN= 000000        CH    = 002000        ECRC  = 001000        EMS10   072074        EMS163  076367
```

EMS164	076413	EMS44	073176	EMT125	066516	EMT207	067420	EMT271	070166		
EMS165	076474	EMS45	073251	EMT126	066526	EMT21	065310	EMT272	070204		
EMS166	076510	EMS46	073314	EMT127	066540	EMT210	067432	EMT273	070222		
EMS167	076535	EMS47	073343	EMT13	065232	EMT211	067440	EMT274	070240		
EMS17	072304	EMS5	071732	EMT130	066552	EMT212	067454	EMT275	070256		
EMS170	076562	EMS50	073372	EMT131	066564	EMT213	067470	EMT276	070270		
EMS171	076611	EMS51	073421	EMT132	066576	EMT214	067500	EMT277	070304		
EMS172	076644	EMS52	073447	EMT133	066610	EMT215	067520	EMT3	065156		
EMS173	076664	EMS53	073457	EMT134	066622	EMT216	067532	EMT30	065400		
EMS174	076701	EMS54	073471	EMT135	066634	EMT217	067542	EMT300	070322		
EMS175	076707	EMS55	073513	EMT136	066646	EMT22	065320	EMT301	070342		
EMS176	076735	EMS56	073542	EMT137	066660	EMT220	067552	EMT302	070364		
EMS177	076764	EMS57	073617	EMT14	065244	EMT221	067562	EMT303	070376		
EMS2	071607	EMS6	071762	EMT140	066670	EMT222	067572	EMT304	070410		
EMS20	072344	EMS60	073645	EMT141	066700	EMT223	067602	EMT305	070422		
EMS200	077006	EMS61	073660	EMT142	066710	EMT224	067612	EMT306	070434		
EMS201	077023	EMS62	073707	EMT143	066720	EMT225	067614	EMT307	070444		
EMS202	077071	EMS63	073725	EMT144	066730	EMT226	067616	EMT31	065410		
EMS203	077121	EMS64	073753	EMT145	066740	EMT227	067620	EMT310	070464		
EMS204	077134	EMS65	073771	EMT146	066750	EMT23	065330	EMT311	070476		
EMS205	077162	EMS66	074037	EMT147	066760	EMT230	067622	EMT312	070510		
EMS206	077174	EMS67	074107	EMT15	065252	EMT231	067624	EMT313	070522		
EMS207	077221	EMS7	072027	EMT150	066770	EMT232	067626	EMT314	070542		
EMS21	072367	EMS70	074134	EMT151	067000	EMT233	067630	EMT315	070554		
EMS210	077246	EMS71	074146	EMT152	067012	EMT234	067632	EMT316	070566		
EMS211	077257	EMS72	074161	EMT153	067024	EMT235	067634	EMT317	070600		
EMS212	077275	EMS73	074171	EMT154	067042	EMT236	067636	EMT32	065420		
EMS213	077320	EMS74	074206	EMT155	067060	EMT237	067640	EMT320	070610		
EMS214	077343	EMS75	074225	EMT156	067072	EMT24	065340	EMT321	070620		
EMS215	077361	EMS76	074233	EMT157	067104	EMT240	067642	EMT322	070630		
EMS216	077402	EMS77	074264	EMT16	065260	EMT241	067644	EMT323	070636		
EMS217	077432	EMTVEC=	000030	EMT160	067116	EMT242	067646	EMT324	070646		
EMS22	072412	EMT1	065144	EMT161	067126	EMT243	067650	EMT325	070660		
EMS220	077465	EMT10	065214	EMT162	067140	EMT244	067652	EMT326	070672		
EMS221	077505	EMT100	066212	EMT163	067152	EMT245	067654	EMT327	070710		
EMS222	077555	EMT101	066230	EMT164	067162	EMT246	067656	EMT33	065430		
EMS223	077606	EMT102	066246	EMT165	067170	EMT247	067666	EMT330	070722		
EMS224	077663	EMT103	066256	EMT166	067176	EMT25	065350	EMT331	070732		
EMS23	072426	EMT104	066270	EMT167	067204	EMT250	067700	EMT332	070744		
EMS24	072454	EMT105	066306	EMT17	065270	EMT251	067714	EMT333	070756		
EMS25	072503	EMT106	066316	EMT170	067212	EMT252	067722	EMT334	070774		
EMS26	072520	EMT107	066326	EMT171	067222	EMT253	067730	EMT335	071012		
EMS27	072541	EMT11	065220	EMT172	067232	EMT254	067746	EMT336	071032		
EMS3	071624	EMT110	066346	EMT173	067242	EMT255	067756	EMT337	071042		
EMS30	072552	EMT111	066360	EMT174	067252	EMT256	067766	EMT34	065440		
EMS31	072562	EMT112	066366	EMT175	067264	EMT257	067776	EMT340	071052		
EMS32	072611	EMT113	066374	EMT176	067272	EMT26	065360	EMT341	071064		
EMS33	072640	EMT114	066410	EMT177	067300	EMT260	070010	EMT342	071072		
EMS34	072666	EMT115	066416	EMT2	065150	EMT261	070022	EMT343	071104		
EMS35	072737	EMT116	066426	EMT20	065300	EMT262	070042	EMT344	071116		
EMS36	072766	EMT117	066436	EMT200	067312	EMT263	070052	EMT345	071130		
EMS37	073015	EMT12	065224	EMT201	067324	EMT264	070062	EMT346	071140		
EMS4	071667	EMT120	066446	EMT202	067336	EMT265	070102	EMT347	071154		
EMS40	073043	EMT121	066456	EMT203	067350	EMT266	070122	EMT35	065450		
EMS41	073072	EMT122	066466	EMT204	067362	EMT267	070134	EMT350	071166		
EMS42	073120	EMT123	066476	EMT205	067400	EMT27	065370	EMT351	071204		
EMS43	073147	EMT124	066506	EMT206	067410	EMT270	070152	EMT352	071214		

SYMBOL TABLE

```
EMT353  071220   FNCMSK= 000077   IR     = 000100   PHA    = 000200   RMDAI   001342
EMT354  071230   F0     = 000002   IVC    = 010000   PIP    = 020000   RMDAO   001416
EMT36   065460   F1     = 000004   LBC    = 002000   PIRQ   = 177772   RMDB   = 000022
EMT37   065470   F2     = 000010   LBT    = 002000   PIRQVE= 000240   RMDBI   001356
EMT4    065164   F3     = 000020   LF     = 000012   PLFS   = 002000   RMDBO   001432
EMT40   065500   F4     = 000040   LODEV    064305   PRIERR  040106   RMDC   = 000034
EMT41   065506   GENBUF  036224   LS     = 000004   PR0    = 000000   RMDCI   001370
EMT42   065516   GET      037110   LSC    = 004000   PR1    = 000040   RMDCO   001444
EMT43   065530   GETBUF  001334   LST    = 000002   PR2    = 000100   RMDS   = 000012
EMT44   065540   GETINX  001522   LSTRK    001332   PR3    = 000140   RMDSI   001346
EMT45   065550   GETSTS  037024   MCLK   = 004000   PR4    = 000200   RMDSO   001422
EMT46   065560   GO     = 000001   MCPE   = 020000   PR5    = 000240   RMDT   = 000026
EMT47   065570   GTSWR  = 104407   MDF    = 000100   PR6    = 000300   RMDTI   001362
EMT5    065172   HCE    = 000200   MDPE   = 000400   PR7    = 000340   RMDTO   001436
EMT50   065576   HCI    = 002000   MEDENB  001510   PS     = 177776   RMEC1  = 000044
EMT51   065606   HCRC   = 000400   MFGFIL  103216   PSEL   = 002000   RMEC1I  001400
EMT52   065620   HELP     101206   MI     = 000004   PSW    = 177776   RMEC1O  001454
EMT53   065636   HT     = 000011   MIXED    064516   PUT      037360   RMEC2  = 000046
EMT54   065654   IAE    = 002000   MOC    = 000400   PUTBUF  001410   RMEC2I  001402
EMT55   065664   IDXMSK= 000077   MOH    = 020000   PUTINX  001551   RMEC2O  001456
EMT56   065676   IE     = 000100   MOL    = 010000   PWRVEC= 000024   RMER1  = 000014
EMT57   065714   ILF    = 000001   MRD    = 002000   QUES     063225   RMER1I  001350
EMT6    065200   ILF02  = 000002   MS     = 000040   RCLSTS  050256   RMER1O  001424
EMT60   065724   ILF24  = 000024   MSC    = 000002   RD     = 000070   RMER2  = 000042
EMT61   065742   ILF26  = 000026   MSDRVS  064263   RDCHR  = 104411   RMER2I  001376
EMT62   065762   ILF30  = 000030   MSE    = 100000   RDLIN  = 104412   RMER2O  001452
EMT63   065776   ILF32  = 000032   MSER   = 000200   RDOCT  = 104413   RMHR   = 000036
EMT64   066000   ILF34  = 000034   MSGDRV  064277   RDY    = 000200   RMHRI   001372
EMT65   066020   ILF36  = 000036   MSHELP  063234   READY    007764   RMHRO   001446
EMT66   066040   ILF40  = 000040   MUR    = 001000   RECAL  = 000006   RMLA   = 000020
EMT67   066052   ILF42  = 000042   MWD    = 000010   RESREG= 104415   RMLAI   001354
EMT7    065206   ILF44  = 000044   MWP    = 000010   RESVEC= 000010   RMLAO   001430
EMT70   066064   ILF46  = 000046   MXF    = 001000   REX    = 010000   RMMR1  = 000024
EMT71   066074   ILF54  = 000054   NDTMSK= 115760   RG     = 040000   RMMR1I  001360
EMT72   066104   ILF56  = 000056   NED    = 010000   RGDTPT  064516   RMMR1O  001434
EMT73   066122   ILF64  = 000064   NEM    = 004000   RH     = 000072   RMMR2  = 000040
EMT74   066140   ILF66  = 000066   NOP    = 000000   RIP    = 000020   RMMR2I  001374
EMT75   066150   ILF74  = 000074   NOTAVL  064342   RLEASE= 000012   RMMR2O  001450
EMT76   066170   ILF76  = 000076   NOTPRS  064325   RMAS   = 000016   RMOF   = 000032
EMT77   066200   ILR    = 000002   NSA    = 100000   RMASI   001352   RMOFI   001366
ENRGDT  065144   ILRG50= 000050   OCC    = 100000   RMASO   001426   RMOFO   001442
EQUALS  063216   ILRG52= 000052   OFD    = 000200   RMBA   = 000004   RMR    = 000004
ERR    = 040000   ILRG54= 000054   OFFLIN  064361   RMBAE  = 000050   RMSN   = 000030
ERRNMB  033306   ILRG56= 000056   OFFSET= 000014   RMBAEI  001404   RMSNI   001364
ERROR  = 104000   ILRG60= 000060   OM     = 000001   RMBAEO  001460   RMSNO   001440
ERRTYP  032550   ILRG62= 000062   ONES     064556   RMBAI   001340   RMWC   = 000002
ERRVEC= 000004   ILRG64= 000064   ONLINE  064373   RMBAO   001414   RMWCI   001336
ERTY00  033314   ILRG66= 00006e   OPE    = 020000   RMCS1  = 000000   RMWCO   001412
ERTY01  033322   ILRG70= 000070   OPI    = 020000   RMCS1I  001334   RQA    = 100000
ERTY02  033332   ILRG72= 000072   OR     = 000200   RMCS1O  001410   RQB    = 040000
ERTY03  033341   ILRG74= 000074   PACACK= 000022   RMCS2  = 000010   RTC    = 000016
ERTY04  033347   ILRG76= 000076   PAKACK= 000022   RMCS2I  001344   R6     =%000006
ESRC   = 004000   IOTVEC= 000020   PAR    = 000010   RMCS2O  001420   R7     =%000007
FER    = 000020   IPCK0  = 000001   PAT    = 000020   RMCS3  = 000052   SADMSK= 000377
FIND   = 000001   IPCK1  = 000002   PDA    = 000400   RMCS3I  001406   SAVREG= 104414
FMT16  = 010000   IPCK2  = 000004   PGE    = 002000   RMCS3O  001462   SA1    = 000001
FNCDTB  064406   IPCK3  = 000010   PGM    = 001000   RMDA   = 000006   SA16   = 000020
```

SYMBOL TABLE

SEQ 0270

```
SA2    = 000002
SA4    = 000004
SA8    = 000010
SC     = 100000
SCOPE  = 000004
SCTMSG  063120
SCTMSK= 003700
SC0    = 000100
SC1    = 000200
SC2    = 000400
SC3    = 001000
SC4    = 002000
SEARCH= 000030
SECERR  040740
SEEK   = 000004
SEKSTS  045224
SHUT    056756
SHUT2   057034
SIZCLK  037600
SKI    = 040000
SNGPRT= 020024
STACK  = 001100
STANDA  006706
START   005420
STCDRV  056222
STKLMT= 177774
STOP    056726
STSD1   101076
STSD2   101114
STSD3   101134
STSD4   101146
STSF    101177
STSH1   100541
STSH2   100616
STSH3   100705
STSH4   100743
SWR     001154
SWREG   000176
SW0    = 000001
SW00   = 000001
SW01   = 000002
SW02   = 000004
SW03   = 000010
SW04   = 000020
SW05   = 000040
SW06   = 000100
SW07   = 000200
SW08   = 000400
SW09   = 001000
SW1    = 000002
SW10   = 002000
SW11   = 004000
SW12   = 010000
SW13   = 020000
SW14   = 040000
SW15   = 100000
SW2    = 000004
SW3    = 000010
SW4    = 000020
SW5    = 000040
SW6    = 000100
SW7    = 000200
SW8    = 000400
SW9    = 001000
TADMSK= 177400
TAG    = 020000
TAGADR= 001114
TAP    = 040000
TA1    = 000400
TA16   = 010000
TA2    = 001000
TA4    = 002000
TA8    = 004000
TBITVE= 000014
TIMOUT  037722
TKVEC  = 000060
TPVEC  = 000064
TRAPVE= 000034
TRE    = 040000
TRTVEC= 000014
TST    = 010000
TSTNMB  033304
TSTPRP  033352
TSTQUE  001464
TST1    010130
TST10   014130
TST11   014750
TST12   015540
TST13   016476
TST14   017272
TST15   020104
TST16   020716
TST17   021540
TST2    010314
TST20   022302
TST21   023044
TST22   023606
TST23   024112
TST24   024416
TST25   024760
TST26   025302
TST27   025620
TST3    010500
TST30   026532
TST31   027354
TST32   030136
TST33   031306
TST4    010702
TST5    011544
TST6    012400
TST7    013170
TYPBN  = 104406
TYPDS  = 104405
TYPE   = 104401
TYPOC  = 104402
TYPON  = 104404
TYPOS  = 104403
UBUSQS  063611
UNS    = 040000
UNTMSK= 000007
UPE    = 020000
USE    = 040000
USRFIL  104224
U0     = 000001
U1     = 000002
U2     = 000004
VV     = 000100
WC     = 000040
WCD    = 000050
WCE    = 040000
WCEHI  = 010000
WCELO  = 004000
WCF    = 000040
WCH    = 000052
WD     = 000060
WH     = 000062
WLE    = 004000
WRL    = 004000
XSIZ    006462
XXDP    001330
ZEROS   064620
$APTHD  001100
$ATYC   062676
$ATY1   062652
$ATY3   062660
$ATY4   062670
$AUTOB  001150
$BASE   001276
$BDADR  001136
$BDDAT  001142
$BELL   001212
$BIN    057212
$CDW1   001302
$CDW2   001304
$CHARC  060216
$CKSWR  061324
$CMTAG  001114
$CM3   = 000000
$CM4   = 000005
$CNTLC  062222
$CNTLG  062234
$CNTLU  062227
$CPUOP  001250
$CRLF   001217
$DBLK   057430
$DDW0   001306
$DDW1   001310
$DDW2   001312
$DDW3   001314
$DDW4   001316
$DDW5   001320
$DDW6   001322
$DDW7   001324
$DEVCT  001232
$DEVM   001300
$DOAGN  032540
$DTBL   057420
$ENDAD  032530
$ENDCT  032374
$ENULL  032544
$ENV    001242
$ENVM   001243
$EOP    032340
$EOPCT  032366
$EOSP   032302
$ERFLG  001117
$ERMAX  001131
$ERROR  060626
$ERRPC  001132
$ERRTB  001600
$ERTTL  001126
$ESCAP  001210
$ETABL  001242
$ETEND  001326
$FATAL  001224
$FFLG   063116
$FILLC  001172
$FILLS  001171
$GDADR  001134
$GDDAT  001140
$GET42  032520
$GTSWR  061414
$HD    = 000000
$HIBTS  001100
$HIOCT  062364
$ICNT   001120
$ILLUP  062634
$INTAG  001151
$ITEMB  001130
$LF     001220
$LFLG   063115
$LPADR  001122
$LPERR  001124
$MADR1  001254
$MADR2  001260
$MADR3  001264
$MADR4  001270
$MAIL   001222
$MAMS1  001252
$MAMS2  001256
$MAMS3  001262
$MAMS4  001266
$MBADR  001102
$MFLG   063114
$MNEW   062252
$MSGAD  001236
$MSGLG  001240
$MSGTY  001222
$MSWR   062241
$MTYP1  001253
$MTYP2  001257
$MTYP3  001263
$MTYP4  001267
$MXCNT  060536
$NULL   001170
$NWTST= 000001
$OCNT   057662
$OMODE  057664
$OVER   060522
$PASS   001230
$PASTM  001106
$POWER  062642
$PWRDN  062474
$PWRMG  062630
$PWRUP  062546
$QUES   001216
$RDCHR  061666
$RDLIN  061756
$RDOCT  062264
$RDSZ  = 000010
$RESRE  057102
$RTNAD  032542
$SAVRE  057044
$SAVR6  062640
$SCOPE  060222
$SETUP= 000137
$STUP  = 177777
$SVLAD  060466
$SVPC  = 000204
$SWR   = 167400
$SWREG  001244
$SWRMK= 000000
$SW08T  060540
$TESTN  001226
$TIMES  001206
$TKB    001162
$TKCNT  061026
$TKINT  061036
$TKQEN= 061035
$TKQIN  061030
$TKQOU  061032
$TKQSR  061034
$TKS    001160
$TKSRV  061106
$TMP0   001174
$TMP1   001176
$TMP2   001200
$TMP3   001202
$TMP4   001204
$TN    = 000034
$TPB    001166
$TPFLG  001173
$TPS    001164
$TRAP   062366
$TRAP2  062426
```

```
$TRP  = 000016          $TYPBN  057140          $TYPOC  057464          $USWR   001246          $XTSTR  060240
$TRPAD  062440          $TYPDS  057214          $TYPON  057500          $VECT1  001272          $$GET4= 000000
$TSTM   001104          $TYPE   057666          $TYPOS  057440          $VECT2  001274          $$SW08= 000034
$TSTNM  001116          $TYPEC  060100          $UNIT   001234          $XOFF = 000023          $OFILL  057663
$TTYIN  062212          $TYPEX  060220          $UNITM  001*10          $XON  = 000021          .$X   = 001100

. ABS.  105232     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  55176 WORDS  ( 216 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  69 PAGES
CZRMNA.BIC,CZRMNA/C=CZRMNA.DOC,CZRMNA,SYSMAC/M
```

$INTAG	S-0#	38-8	38-8	38-8	38-8	38-8*

SEQ 0274

```
$ITEMB   5-0#    14-29    38-7     38-7     38-7     38-7*
$LF      5-0#    38-5     38-5     38-7     38-7     38-8     38-8     38-8
$LFLG   38-12#   38-12*
$LPADR   5-0#     9-19*   38-6     38-6     38-6     38-6*    38-6*    38-6*
$LPERR   5-0#     9-19*   38-6     38-6     38-6     38-6*    38-7
$MADR1   5-0#
$MADR2   5-0#
$MADR3   5-0#
$MADR4   5-0#
$MAIL    4-789    4-789    5-0#     9-19     9-25     11-27    12-2     12-34    12-63    12-91    12-193   12-291   12-384   12-509
        12-605   12-698   12-822   12-915   12-:10   12-;06   12-<06   12-<89   12-=72   12->54   12-?00   12-?44   12-?99   12-@49
        12-@98   12-B05   12-C03   12-C93   12-E56   38-5     38-6     38-7
$MAMS1   5-0#
$MAMS2   5-0#
$MAMS3   5-0#
$MAMS4   5-0#
$MBADR   4-789#
$MFLG   38-12    38-12#   38-12*   38-12*
$MNEW   38-8     38-8#
$MSGAD   5-0#    38-12    38-12*
$MSGLG   5-0#    38-12*
$MSGTY   5-0#    38-12    38-12    38-12*   38-12*
$MSWR   38-8     38-8#
$MTYP1   5-0#
$MTYP2   5-0#
$MTYP3   5-0#
$MTYP4   5-0#
$MXCNT  38-6     38-6     38-6     38-6#
$NULL    5-0#    38-5     38-5     38-5
$NWTST   12-2    12-2#    12-2#    12-34    12-34#   12-34#   12-63    12-63#   12-63#   12-91    12-91#   12-91#   12-193   12-193#
        12-193#  12-291   12-291#  12-291#  12-384   12-384#  12-384#  12-509   12-509#  12-509#  12-605   12-605#  12-605#  12-698
        12-698#  12-698#  12-822   12-822#  12-822#  12-915   12-915#  12-915#  12-:10   12-:10#  12-:10#  12-;06   12-;06#  12-;06#
        12-<06   12-<06#  12-<06#  12-<89   12-<89#  12-<89#  12-=72   12-=72#  12-=72#  12->54   12->54#  12->54#  12-?00   12-?00#
        12-?00#  12-?44   12-?44#  12-?44#  12-?99   12-?99#  12-?99#  12-@49   12-@49#  12-@49#  12-@98   12-@98#  12-@98#  12-B05
        12-B05#  12-B05#  12-C03   12-C03#  12-C03#  12-C93   12-C93#  12-C93#  12-E56   12-E56#  12-E56#
$OCNT   38-4#    38-4*    38-4*
$OMODE  38-4     38-4#    38-4*    38-4*    38-4*    38-4*
$OVER   38-6     38-6     38-6     38-6     38-6#
$PASS    5-0#     9-19*   13-20    13-20    13-20    13-20*   13-20*   38-6     38-6     38-6
$PASTM   4-789#
$POWER  38-11    38-11#
$PWRDN   9-19    38-11    38-11#
$PWRMG  38-11#
$PWRUP  38-11    38-11#
$QUES    5-0#    38-5     38-5     38-7     38-7     38-8     38-8     38-8     38-8
$R2A    38-10
$RDCHR  38-8#    38-10    38-10
$RDDEC  38-10
$RDLIN  38-8#    38-10    38-10
$RDOCT  38-9#    38-10    38-10
$RDSZ   38-8     38-8#
$RESRE  38-1#    38-10
$RTNAD  13-20#
$SAVR6  38-11    38-11#   38-11*   38-11*   38-11*
$SAVRE  38-1#    38-10    38-10
$SCOPE   9-19    38-6#
```

```
$SETUP  4-782   4-782   4-782   4-782   4-782   4-782   4-782#  4-782#  4-782#  4-782#  4-782#  4-782#  4-782#  9-19
                                                                F 6                                             SEQ 0276
```

	9-19	9-19	9-19	9-19	9-19	9-19	9-19	9-19	9-19	9-19	9-19	9-25	9-25	9-25
	11-27	13-20	13-20	38-6	38-7	38-7	38-7	38-7	38-8	38-8	38-8	38-8	38-8	
$STUP	4-782	4-782	4-782	4-782	4-782	4-782	4-782#	4-782#	4-782#	4-782#	4-782#	4-782#	4-782#	4-782#
	4-782#	4-782#	4-782#	4-782#										
$SVLAD	38-6	38-6#												
$SVPC	4-786	4-786#												
$SWOBT	38-6	38-6#												
$SWR	4-476#	4-487	4-488	4-488	4-488	4-488	4-488	4-488	4-488	4-488	5-0	5-0	5-0	9-19
	9-19	9-19	9-19	9-19	12-2	12-34	12-63	12-91	12-193	12-291	12-384	12-509	12-605	12-698
	12-822	12-915	12-:10	12-;06	12-<06	12-<89	12-=72	12->54	12-?00	12-?44	12-?99	12-@49	12-@98	12-B05
	12-C03	12-C93	12-E56	13-20	13-20	13-20	13-20	13-20	38-6	38-6	38-6	38-6	38-6	38-6
	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6	38-6
	38-7	38-7	38-7	38-7	38-7	38-7	38-7	38-7	38-7	38-7	38-7	38-11		
$SWREG	5-0#	9-19												
$SWRMK	4-488	4-488	4-488	4-488	4-488	4-488	4-488	4-488	4-488	38-6	38-6	38-6	38-6	38-6
	38-6	38-6	38-6	38-6	38-6									
$TESTN	5-0#	12-2*	12-34*	12-63*	12-91*	12-193*	12-291*	12-384*	12-509*	12-605*	12-698*	12-822*	12-915*	12-:10*
	12-;06*	12-<06*	12-<89*	12-=72*	12->54*	12-?00*	12-?44*	12-?99*	12-@49*	12-@98*	12-B05*	12-C03*	12-C93*	12-E56*
	14-25	38-6*												
$TIMES	5-0#	9-19*	11-38*	13-20*	38-6	38-6	38-6	38-6*	38-6*					
$TKB	5-0#	38-5	38-5	38-8	38-8	38-8	38-8	38-8	38-8	38-8				
$TKCNT	38-8	38-8	38-8#	38-8*	38-8*	38-8*								
$TKINT	10-4	11-40	38-8	38-8	38-8#									
$TKQEN	38-8	38-8	38-8#											
$TKQIN	38-8	38-8	38-8#	38-8*	38-8*	38-8*	38-8*							
$TKQOU	38-8	38-8	38-8#	38-8*	38-8*	38-8*								
$TKQSR	38-8	38-8	38-8	38-8#										
$TKS	5-0#	38-5	38-5	38-8	38-8	38-8	38-8	38-8	38-8*	38-8*	38-8*	38-8*	38-8*	38-8*
$TKSRV	38-8	38-8#												
$TMP0	5-0#	12-44*	12-47	12-104*	12-205*	12-303*	12-396*	12-521*	12-617*	12-710*	12-834*	12-930*	12-:25*	12-;18*
	12-<19*	12-=02*	12-=85*	12-A10*	12-B17*	12-C16*	12-D08*	12-E10*	12-E71*	12-F43*	17-55	18-85*	20-41*	21-35*
	28-29*	28-31	49-3	49-6	49-10									
$TMP1	5-0#	10-13*	10-14	10-15	10-20	10-29*	10-30	10-31	10-33	10-44*	10-45	10-46	10-48	10-59*
	10-61	10-65	10-75*	10-77	10-81	10-96*	10-98	10-102*	10-103*	10-104*	10-105*	10-106*	10-107	10-118*
	10-119	10-125	10-127	10-128	10-130	10-135*	10-136	10-139	10-140	10-142	10-147	12-43*	12-51	12-84*
	12-105*	12-206*	12-304*	12-397*	12-522*	12-618*	12-711*	12-835*	12-931*	12-:26*	12-;19*	12-<20*	12-=03*	12-=86*
	12-A11*	12-B18*	12-C17*	12-D09*	12-E11*	12-E72*	12-F44*	17-56	20-42*	20-43	21-36*	21-37	27-24*	27-25*
	27-37	27-39	28-28*	28-37	49-3	49-5	49-10							
$TMP2	5-0#	12-85*	27-27*	27-28*	27-50	27-53	49-5							
$TMP3	5-0#													
$TMP4	5-0#													
$TN	4-477#	4-487	12-2	12-2	12-2	12-2#	12-34	12-34	12-34	12-34#	12-63	12-63	12-63	12-63#
	12-91	12-91	12-91	12-91#	12-193	12-193	12-193	12-193#	12-291	12-291	12-291	12-291#	12-384	12-384
	12-384	12-384#	12-509	12-509	12-509	12-509#	12-605	12-605	12-605	12-605#	12-698	12-698	12-698	12-698#
	12-822	12-822	12-822	12-822#	12-915	12-915	12-915	12-915#	12-:10	12-:10	12-:10	12-:10#	12-;06	12-;06
	12-;06	12-;06#	12-<06	12-<06	12-<06	12-<06#	12-<89	12-<89	12-<89	12-<89#	12-=72	12-=72	12-=72	12-=72#
	12->54	12->54	12->54	12->54#	12-?00	12-?00	12-?00	12-?00#	12-?44	12-?44	12-?44	12-?44#	12-?99	12-?99
	12-?99	12-?99#	12-@49	12-@49	12-@49	12-@49#	12-@98	12-@98	12-@98	12-@98#	12-B05	12-B05	12-B05	12-B05#
	12-C03	12-C03	12-C03	12-C03#	12-C93	12-C93	12-C93	12-C93#	12-E56	12-E56	12-E56	12-E56#	38-6	38-6
$TPB	5-0#	38-5	38-5	38-5*										
$TPFLG	5-0#	38-5	38-5	38-5										
$TPS	5-0#	38-5	38-5	38-5										
$TRAP	9-19	38-10#												
$TRAP2	38-10	38-10#												
$TRP	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10
	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10	38-10

38-10 38-10 38-10 38-10 38-10 38-10 38-10 H 6 38-10 38-10 38-10 38-10 38-10 38-10 38-10 SEQ 0278

AENVM 5-0 5-0

J 6

SEQ 0280

CROSS REFERENCE TABLE (CREF V01-05) SEQ 0281

```
AFATAL   5-0       5-0
ALL      10-121    39-6#
AMADR1   5-0       5-0
AMADR2   5-0       5-0
AMADR3   5-0       5-0
AMADR4   5-0       5-0
AMAMS1   5-0       5-0
AMAMS2   5-0       5-0
AMAMS3   5-0       5-0
AMAMS4   5-0       5-0
AMSGAD   5-0       5-0
AMSGLG   5-0       5-0
AMSGTY   5-0       5-0
AMTYP1   5-0       5-0
AMTYP2   5-0       5-0
AMTYP3   5-0       5-0
AMTYP4   5-0       5-0
AOE      4-579#    4-590     25-311    25-326    25-341    25-356    25-360    26-117    26-121    40-77     40-78     40-81     40-82     40-85
         40-86
APASS    5-0       5-0
APE      4-757#
APRIOR   5-0
APTCSU   38-5      38-12#
APTENV   38-5      38-7      38-12     38-12#
APTSIZ   9-19      38-12#
APTSPO   38-5      38-12     38-12#
ARGS     12-102#   12-110    12-110#   12-122#   12-131#   12-135#   12-146#   12-152#   12-156#   12-158#   12-160#   12-167#   12-173#   12-177#
         12-179#   12-181#   12-185#   12-203#   12-211    12-211#   12-223#   12-232#   12-236#   12-248#   12-254#   12-258#   12-260#   12-262#
         12-270#   12-276#   12-280#   12-282#   12-284#   12-288#   12-301#   12-309    12-309#   12-321#   12-330#   12-334#   12-345#   12-351#
         12-355#   12-357#   12-359#   12-365#   12-371#   12-375#   12-377#   12-379#   12-394#   12-402    12-402#   12-414#   12-423#   12-427#
         12-438#   12-444#   12-448#   12-450#   12-452#   12-461#   12-467#   12-471#   12-478#   12-496#   12-519#   12-527    12-527#   12-539#
         12-548#   12-552#   12-563#   12-569#   12-573#   12-575#   12-577#   12-584#   12-590#   12-594#   12-596#   12-598#   12-602#   12-615#
         12-623    12-623#   12-635#   12-644#   12-648#   12-659#   12-665#   12-669#   12-671#   12-673#   12-679#   12-685#   12-689#   12-691#
         12-693#   12-708#   12-716    12-716#   12-728#   12-737#   12-741#   12-752#   12-758#   12-762#   12-764#   12-766#   12-775#   12-781#
         12-785#   12-792#   12-808#   12-832#   12-840    12-840#   12-852#   12-861#   12-865#   12-876#   12-882#   12-886#   12-888#   12-890#
         12-896#   12-902#   12-906#   12-908#   12-910#   12-926#   12-935    12-935#   12-947#   12-956#   12-960#   12-971#   12-977#   12-981#
         12-983#   12-985#   12-991#   12-997#   12-:01#   12-:03#   12-:05#   12-:21#   12-:31    12-:31#   12-:43#   12-:52#   12-:56#   12-:67#
         12-:73#   12-:77#   12-:79#   12-:81#   12-:87#   12-:93#   12-:97#   12-:99#   12-;01#   12-;16#   12-;24    12-;24#   12-;38#   12-;47#
         12-;51#   12-;63#   12-;69#   12-;73#   12-;75#   12-;77#   12-;83#   12-;89#   12-;93#   12-;95#   12-;97#   12-<17#   12-<25    12-<25#
         12-<39#   12-<46#   12-<50#   12-<52#   12-<54#   12-<61#   12-<67#   12-<71#   12-<73#   12-<75#   12-<79#   12-=00#   12-=08    12-=08#
         12-=22#   12-=29#   12-=33#   12-=35#   12-=37#   12-=44#   12-=50#   12-=54#   12-=56#   12-=58#   12-=62#   12-=83#   12-=92    12-=92#
         12->05#   12->13#   12->17#   12->19#   12->21#   12->27#   12->32#   12->36#   12->38#   12->40#   12->44#   12->67    12->67#   12->80#
         12->87#   12->91#   12->93#   12->95#   12-?13    12-?13#   12-?24#   12-?31#   12-?35#   12-?37#   12-?39#   12-?55    12-?55#   12-?68#
         12-?75#   12-?79#   12-?81#   12-?83#   12-@11    12-@11#   12-@24#   12-@31#   12-@35#   12-@37#   12-@39#   12-@60    12-@60#   12-@73#
         12-@80#   12-@84#   12-@86#   12-@88#   12-A08#   12-A16    12-A16#   12-A28#   12-A39#   12-A46#   12-A50#   12-A52#   12-A54#   12-A65#
         12-A78#   12-A84#   12-A88#   12-A90#   12-A92#   12-A96#   12-B15#   12-B23    12-B23#   12-B31#   12-B47#   12-B54#   12-B58#   12-B67#
         12-B71    12-B71#   12-B78#   12-B85#   12-B89#   12-B98#   12-C14#   12-C22    12-C22#   12-C36#   12-C43#   12-C47#   12-C49#   12-C51#
         12-C61#   12-C67#   12-C71#   12-C78#   12-C85#   12-D06#   12-D14    12-D14#   12-D36#   12-D43#   12-D47#   12-D49#   12-D51#   12-D58#
         12-D64#   12-D68#   12-D79#   12-D92#   12-E05#   12-E17#   12-E38    12-E38#   12-E49#   12-E69#   12-E77    12-E77#   12-E99#   12-F06#
         12-F10#   12-F12#   12-F14#   12-F21#   12-F27#   12-F31#   12-F38#   12-F50#   12-F71    12-F71#   12-F82#
ASNDA    6-0#      16-300*   16-332    16-359*   16-360    16-362*   16-363*   16-364    16-366*   16-398
ASNDC    6-0#      16-299*   16-331    16-367*   16-368    16-370*   16-397
ASWREG   5-0       5-0
ATA      4-559#    25-138    25-139    25-141    29-156    29-157    29-192    29-195    33-145    33-146    33-185    33-188    40-58     40-59
         40-60     40-63     40-64     40-67     40-68     40-69     40-70     40-71     40-72     40-73     40-74     40-75     40-76     40-79
         40-80     40-83     40-84     40-87     40-88
```

ATESTN 5-0 5-0

L 6

SEQ 0282

12-D79 12-D79# 12-D79# 12-D92 12-D92# 12-D92# 12-E05 N 6 12-E05# 12-E05# 12-E17 12-E17# 12-E17# 12-E38 12-E38#

SEQ 0284

```
                12-E49   12-E49#  12-E69   12-E69#  12-E77   12-E77#  12-E99   12-E99#  12-F06   12-F06#  12-F10   12-F10#  12-F10#  12-F12
                12-F12#  12-F12#  12-F14   12-F14#  12-F14#  12-F21   12-F21#  12-F27   12-F27#  12-F31   12-F31#  12-F31#  12-F38   12-F38#
                12-F38#  12-F50   12-F50#  12-F50#  12-F71   12-F71#  12-F82   12-F82#
BADSCT          12-102   12-203   12-301   12-394   12-519   12-615    2-708   12-832   12-926   12-:21   12-;16   12-<17   12-=00   12-=83
                12-A08   12-B15   12-C14   12-D06   12-E69   16-33#
BADTMO           9-3#     9-21
BA1             4-746#   26-28
BB00            4-677#
BB01            4-677#
BB02            4-677#
BB03            4-677#
BB04            4-677#
BB05            4-677#
BB06            4-677#
BB07            4-677#
BB08            4-677#
BB09            4-677#
BIT0            4-491#   12-74    12-76    12-79    12-81    23-18
BIT00           4-491     4-491#   4-503    4-551    4-569    4-588    4-625    4-643    4-677    4-749    4-767
BIT01           4-491     4-491#   4-502    4-550    4-587    4-624    4-642    4-677    4-749    4-766
BIT02           4-491     4-491#   4-501    4-549    4-586    4-623    4-641    4-677    4-749    4-765
BIT03           4-491     4-491#   4-500    4-548    4-585    4-622    4-640    4-677    4-688    4-746    4-764
BIT04           4-491     4-491#   4-499    4-547    4-584    4-639    4-677    4-745
BIT05           4-491     4-491#   4-498    4-583    4-621    4-638    4-677    4-744
BIT06           4-491     4-491#   4-568    4-582    4-604    4-620    4-637    4-677    4-730    4-743    4-763
BIT07           4-491     4-491#   4-567    4-581    4-603    4-619    4-636    4-660    4-677    4-687    4-729    4-742
BIT08           4-491     4-491#   4-544    4-566    4-580    4-602    4-618    4-635    4-677    4-728    4-741    38-6
BIT09           4-491     4-491#   4-543    4-565    4-579    4-601    4-617    4-634    4-677    4-727    4-740    38-6     38-7
BIT1            4-491#   12-79    12-81    16-305   17-26    26-48
BIT10           4-491#    4-542    4-564    4-578    4-600    4-616    4-633    4-659    4-674    4-686    4-726    4-739    4-762    38-7
BIT11           4-491#    4-497    4-541    4-563    4-577    4-615    4-632    4-650    4-658    4-673    4-685    4-738    4-761    15-165
                38-6
BIT12           4-491#    4-540    4-562    4-576    4-614    4-631    4-657    4-672    4-684    4-737    4-760    15-111
BIT13           4-491#    4-561    4-575    4-613    4-630    4-649    4-671    4-683    4-725    4-736    4-759    38-7
BIT14           4-491#    4-560    4-574    4-612    4-629    4-648    4-670    4-682    4-693    4-724    4-735    4-758    15-73    38-6
BIT15           4-491#    4-559    4-573    4-611    4-628    4-647    4-669    4-681    4-692    4-723    4-734    4-757    18-33    18-90
BIT2            4-491#   23-18
BIT3            4-491#   15-196   35-320   35-407
BIT4            4-491#   15-141   35-426
BIT5            4-491#
BIT6            4-491#   15-87
BIT7            4-491#    9-68    23-24
BIT8            4-491#
BIT9            4-491#
BOTADR          14-53*   14-71*   14-74    14-89    14-134#
BOTFLG          14-39*   14-81*   14-84    14-87*   14-135#
BPTVEC          4-491#
BSE             4-681#   12-D89   12-D96   16-221   25-431
BUFFER          16-49*   16-285   51-2#    51-16
BUFONE          12-99    12-185   12-200   12-288   12-298   12-39i   12-516   12-602   12-612   12-705   12-829   12-923   12-:18   12-;13
                12-<14   12-<79   12-<97   12-=62   12-=80   12->44   12-?49   12-@06   12-@55   12-A05   12-A96   12-B01   12-B12   12-C11
                12-D03   12-D18   12-D20*  12-D22*  12-E34   12-E66   12-E81   12-E83*  12-E85*  12-F67   51-3#
BUFTWO          12-165   12-185   12-268   12-288   12-456*  12-488   12-498   12-582   12-602   12-770*  12-800   12-810   12-<59   12-<79
                12-=42   12-=62   12->26   12->44   12->63   12-?09   12-A71   12-A96   12-B76   12-C58   12-D56   12-F19   51-4#
CC              4-673#
CH              4-674#
```

CHRCNT	14-40*	14-58*	14-64*	14-65	14-68*	14-72	14-77*	14-88*	14-136#

C 7

SEQ 0286

```
CKSWR    38-6     38-7     38-7     38-10#
CLKADR    6-0#    22-8*    22-10    22-14*   22-16    23-13
CLKVCT    6-0#    22-9*    22-15*
CLR       4-744#   9-82    11-49    30-16
CLRSTS   15-98    31-24#
CMNSTA    9-114   10-18    11-2#
CMPBUF   12-185   12-288   12-602   12-<79   12-=62   12->44   12-A96   18-17#
CMPERR   27-21#
CNSL00   10-11    39-16#
CNSL01   10-55    39-17#
CNSL02   10-63    39-18#
CNSL03   10-67    39-19#
CNSL04   10-79    39-20#
CNSL05   10-83    39-21#
CNSL06   10-100   39-22#
CNSL07   10-113   39-23#
CNSL08   10-22    10-35    10-50    10-144   39-26#
CNSL09   39-27#
CNTCLR   12-36    12-65    15-76    30-12#
COMMA    10-138   39-8#
CONT      4-637#
CR        4-491#  10-125   10-136   14-56    38-5     38-5     47-79
CRLF      4-491#   9-6      9-25     9-25     9-39     9-50     9-56     9-57     11-24    37-18    37-18    38-5     38-5     39-3
          39-4     39-6     39-9     39-10    39-11    39-12    39-12    39-13    39-15    39-16    39-17    39-18    39-20    39-22
          39-23    39-24    39-25    39-26    39-27    39-28    48-7     48-11    48-12    48-15    48-16    48-18    51-19    51-20
          51-21    51-22    51-23    51-24    51-25    51-26    51-27    51-28    51-29    51-30    51-31    51-32    51-33    51-34
          51-35    51-36    51-37    51-38    51-39    51-40    51-41    51-42    51-43    51-44    51-45    51-46    51-47    51-48
          51-49    51-50    51-51    51-52    51-53    51-54    51-55    51-56    51-57    51-58    51-59    51-60    51-61    51-62
          51-63    51-64    51-65    51-66    51-67    51-68    51-69    51-70
CTLFG     6-0#    11-39*   37-13    37-22*
CYLMSK    4-664#
DBCK      4-611#
DBEN      4-612#
DBL       4-762#
DCK       4-573#   4-590    16-225   18-23    25-469   25-489   35-413   35-416
DDISP     4-491#   5-0       9-19
DEBL      4-613#
DEVSEL   15-62    28-12#
DISPLA    5-0#     9-19*    9-19*    38-6*    38-7*
DISPRE    4-784#   9-19
DLT       4-734#  25-313   35-492   35-495
DMD       4-625#   4-643#  12-B27
DPE       4-688#  24-139   25-255   29-32    33-31    35-43    3_-466   35-469
DPEHI     4-758#
DPELO     4-759#
DPR       4-566#  29-195   31-127   31-128   34-30
DRQ       4-650#
DRVCLR    4-511#  16-87    34-18
DRVSTS   16-105   34-9#
DRY       4-567#  24-106   25-38    25-41    25-42    29-195   31-127   31-128   34-30
DSWR      4-491#   5-0       9-19
DTASTS   12-158   12-179   12-260   12-282   12-357   12-377   12-450   12-478   12-575   12-596   12-671   12-691   12-764   12-792
         12-888   12-908   12-983   12-:03   12-:79   12-:99   12-;75   12-;95   12-<52   12-<73   12-=35   12- 56   12->19   12->38
         12->93   12-?37   12-?81   12-@37   12-@86   12-A52   12-A90   12-C49   12-C78   12-D49   12-D79   12-D92   12-E05   12-F12
         12-F38   16-200   35-17#
DTE       4-576#   4-590   35-279   35-297   35-300
```

7-526 7-532 7-535 7-538 7-541 7-544 7-547 7-550 7-553 7-557 7-560 7-564 7-567 7-570

G 7

SEQ 0290

	7-574	7-578	7-582	7-587	7-591	7-595	7-598	7-601	7-604	7-607	7-610	7-613	7-616	7-619
	7-622	7-625	7-628	7-631	7-634	7-637	7-640	7-643	7-646	7-649	7-652	7-655	7-658	7-661
	7-664	7-667	7-670	7-673	7-676	7-679	7-694	7-697	7-703	7-706	7-709	7-712	7-715	7-724
	46-1#													
EFT110	7-222	46-4#												
EFT111	7-225	7-228	46-5#											
EFT114	7-234	46-7#												
EFT2	7-427	7-430	7-514	7-517	46-2#									
EFT223	7-448	7-529	46-8#											
EFT336	7-352	7-682	7-688	7-691	46-10#									
EFT337	7-685	46-11#												
EFT344	7-700	46-12#												
EFT353	7-721	46-14#												
EH1	44-1	48-1#												
EH110	44-4	48-2#												
EH111	44-5	48-3#												
EH114	44-7	48-5#												
EH223	44-8	48-6#												
EH256	44-9	48-7#												
EH336	44-11	48-10#												
EH337	44-12	48-11#												
EH344	44-13	48-15#												
EH353	44-15	48-18#												
EHT1	7-4	7-7	7-10	7-13	7-16	7-19	7-25	7-28	7-31	7-34	7-37	7-40	7-43	7-46
	7-49	7-52	7-55	7-58	7-61	7-64	7-67	7-70	7-73	7-76	7-79	7-82	7-85	7-88
	7-91	7-94	7-97	7-100	7-103	7-106	7-109	7-112	7-115	7-118	7-121	7-124	7-127	7-131
	7-134	7-137	7-140	7-143	7-147	7-151	7-155	7-161	7-164	7-167	7-170	7-173	7-176	7-179
	7-183	7-186	7-189	7-192	7-195	7-198	7-201	7-204	7-207	7-210	7-213	7-216	7-219	7-237
	7-240	7-243	7-246	7-249	7-252	7-255	7-258	7-261	7-264	7-267	7-270	7-273	7-276	7-279
	7-282	7-285	7-288	7-291	7-294	7-297	7-300	7-303	7-306	7-309	7-312	7-315	7-318	7-321
	7-324	7-327	7-330	7-333	7-336	7-339	7-342	7-345	7-348	7-355	7-358	7-361	7-364	7-370
	7-373	7-37(	7-379	7-382	7-385	7-388	7-391	7-394	7-397	7-400	7-403	7-406	7-409	7-412
	7-415	7-41(	7-421	7-424	7-433	7-436	7-439	7-442	7-445	7-505	7-508	7-511	7-520	7-523
	7-526	7-532	7-535	7-538	7-541	7-544	7-547	7-550	7-553	7-557	7-560	7-564	7-567	7-570
	7-574	7-578	7-582	7-587	7-591	7-595	7-598	7-601	7-604	7-607	7-610	7-613	7-616	7-619
	7-622	7-625	7-628	7-631	7-634	7-637	7-640	7-643	7-646	7-649	7-652	7-655	7-658	7-661
	7-664	7-667	7-670	7-673	7-676	7-679	7-694	7-697	7-703	7-706	7-709	7-712	7-715	7-724
	44-1#													
EHT110	7-222	44-4#												
EHT111	7-225	7-228	44-5#											
EHT114	7-234	44-7#												
EHT2	7-427	7-430	7-514	7-517	44-2#									
EHT223	7-448	44-8#												
EHT256	7-529	44-9#												
EHT336	7-352	7-682	7-688	7-691	44-11#									
EHT337	7-685	44-12#												
EHT344	7-700	44-13#												
EHT353	7-721	44-15#												
EMS1	43-3	47-3#												
EMS10	43-7	43-11	47-10#											
EMS100	43-39	43-71	43-176	43-196	47-66#									
EMS101	43-40	43-59	43-174	43-197	47-67#									
EMS102	43-177	43-178	43-181	43-182	43-183	43-184	43-186	43-187	43-188	43-189	43-234	43-235	47-68#	
EMS103	43-56	43-57	43-135	43-141	43-196	43-197	43-198	43-203	43-204	43-205	43-206	43-207	43-208	43-209
	43-210	43-211	43-213	43-214	43-215	43-219	43-220	47-69#						
EMS104	43-136	43-180	43-203	47-70#										

EMS105 43-204 47-71#

1 7

SEQ 0292

EMS106	43-212	43-232	47-72#											
EMS11	43-12	43-15	43-16	43-17	43-18	43-19	43-20	43-21	43-22	43-23	43-24	43-25	43-26	43-27
	43-28	43-29	43-30	43-31	43-32	43-33	43-43	43-177	47-11#					
EMS110	43-9	43-74	43-77	47-73#										
EMS111	43-74	43-226	47-74#											
EMS112	43-74	47-75#												
EMS113	43-75	43-76	43-78	43-127	47-77#									
EMS114	43-77	47-78#												
EMS115	43-44	43-45	43-48	43-50	43-51	43-52	43-54	43-55	43-60	43-61	43-63	43-66	43-67	43-70
	43-73	43-77	43-107	43-108	43-109	43-110	43-134	43-137	43-139	43-140	43-142	43-179	43-182	43-183
	43-185	43-186	43-187	43-188	43-189	43-190	43-192	43-193	43-194	43-195	43-201	43-205	43-221	43-222
	43-223	43-232	43-234	47-79#										
EMS116	43-74	43-77	43-191	43-225	47-80#									
EMS117	43-77	43-143	43-148	43-149	43-178	43-185	43-201	43-225	43-228	43-229	43-230	43-231	47-81#	
EMS12	43-12	47-12#												
EMS120	43-78	47-82#												
EMS121	43-79	43-88	43-97	43-99	47-83#									
EMS122	43-80	43-89	47-84#											
EMS123	43-81	43-88	43-89	43-90	43-90	43-91	43-92	43-93	43-94	43-95	43-96	43-98	43-126	47-85#
EMS124	43-56	43-82	43-91	43-101	47-86#									
EMS125	43-83	43-92	43-102	43-169	47-87#									
EMS126	43-84	43-93	43-103	47-88#										
EMS127	47-89#													
EMS13	43-13	43-42	43-216	47-13#										
EMS130	43-86	43-95	43-106	47-90#										
EMS131	43-87	43-96	43-104	47-91#										
EMS132	43-79	43-80	43-81	43-82	43-83	43-84	43-85	43-86	43-87	43-88	43-89	43-90	43-91	43-92
	43-93	43-94	43-95	43-96	43-97	43-98	43-99	43-100	43-101	43-102	43-103	43-104	43-105	43-106
	43-125	43-126	43-168	43-169	47-92#									
EMS133	43-79	43-80	43-81	43-82	43-83	43-84	43-85	43-86	43-87	43-125	47-93#			
EMS134	43-88	43-89	43-90	43-91	43-92	43-93	43-94	43-95	43-96	43-126	47-94#			
EMS135	43-97	43-98	47-95#											
EMS136	43-99	43-100	43-101	43-102	43-103	43-104	43-105	43-106	47-96#					
EMS137	43-85	43-94	43-100	47-97#										
EMS14	43-13	47-14#												
EMS140	43-44	43-52	43-60	43-107	43-117	43-137	43-186	43-192	47-98#					
EMS141	43-35	43-44	43-45	43-60	43-61	43-66	43-107	43-129	43-137	43-187	43-188	43-193	43-222	47-99#
EMS142	43-47	43-65	43-108	43-140	43-184	43-202	47-100#							
EMS143	43-47	43-65	43-73	43-108	43-109	43-110	43-133	43-140	43-184	43-202	43-205	47-101#		
EMS144	43-36	43-73	43-109	43-110	43-205	47-102#								
EMS145	43-36	43-73	43-109	43-110	43-205	47-103#								
EMS146	43-109	43-185	43-190	43-201	47-104#									
EMS147	43-69	43-112	43-113	43-173	43-217	47-105#								
EMS15	43-13	43-42	43-216	47-15#										
EMS150	43-50	43-51	43-54	43-63	43-67	43-111	43-134	43-138	43-182	43-183	43-194	43-195	43-223	43-234
	47-106#													
EMS151	43-69	43-112	43-113	43-173	43-217	47-107#								
EMS152	43-50	43-51	43-54	43-63	43-111	43-134	43-182	43-183	43-194	43-195	43-223	47-108#		
EMS153	43-111	47-109#												
EMS154	43-111	43-112	43-122	43-123	43-124	47-110#								
EMS155	43-112	43-124	47-111#											
EMS156	43-113	43-114	43-115	43-116	43-173	47-112#								
EMS157	43-113	43-123	47-113#											
EMS16	43-13	47-16#												
EMS160	43-114	43-115	43-116	47-114#										
EMS161	43-114	47-115#												

EMS163	43-58	43-59	43-60	43-61	43-62	43-63	43-64	43-65	43-66	43-67	43-68	43-69	43-70	43-71
	43-72	43-73	43-117	43-143	47-117#									
EMS164	43-45	47-118#												
EMS165	43-56	43-57	43-71	43-72	43-172	43-196	43-197	43-198	47-120#					
EMS166	43-122	43-142	47-121#											
EMS167	43-8	43-49	43-68	43-122	43-123	43-124	43-168	43-169	43-170	43-175	43-235	47-122#		
EMS17	43-14	47-17#												
EMS170	43-9	47-123#												
EMS171	43-57	43-105	43-125	43-126	47-124#									
EMS172	43-215	47-125#												
EMS173	43-214	47-126#												
EMS174	43-216	47-127#												
EMS175	43-216	43-217	43-218	43-219	47-128#									
EMS176	47-129#													
EMS177	43-127	47-130#												
EMS2	43-4	43-5	47-4#											
EMS20	43-34	43-58	43-110	43-118	43-119	43-120	43-121	43-212	43-232	43-238	47-18#			
EMS200	43-128	47-131#												
EMS201	43-128	47-132#												
EMS202	43-129	43-130	43-131	43-132	43-133	43-134	43-136	43-180	47-133#					
EMS203	43-129	43-130	43-131	43-132	43-133	43-174	43-175	43-176	43-237	47-134#				
EMS204	43-131	47-135#												
EMS205	43-55	43-142	43-179	43-233	47-136#									
EMS206	43-55	43-142	43-170	43-179	43-233	47-137#								
EMS207	43-168	43-170	47-138#											
EMS21	43-15	47-19#												
EMS210	43-169	43-170	43-173	47-139#										
EMS211	43-170	47-140#												
EMS212	43-171	43-172	47-141#											
EMS213	43-171	47-142#												
EMS214	43-55	47-143#												
EMS215	43-236	47-144#												
EMS216	43-237	47-145#												
EMS217	43-237	47-146#												
EMS22	43-16	47-20#												
EMS220	43-179	47-147#												
EMS221	43-231	43-238	47-148#											
EMS222	43-233	47-149#												
EMS223	43-232	47-150#												
EMS224	43-121	47-151#												
EMS23	43-17	47-21#												
EMS24	43-18	47-22#												
EMS25	43-19	43-115	47-23#											
EMS26	43-20	43-116	43-173	47-24#										
EMS27	43-16	43-19	43-20	43-115	43-116	43-173	47-25#							
EMS3	43-4	43-76	47-5#											
EMS30	43-17	43-18	43-21	43-22	43-23	43-24	43-25	43-26	43-27	43-28	43-29	43-30	43-31	43-32
	43-33	43-43	43-129	43-130	43-131	43-132	43-133	43-144	43-145	43-146	43-147	43-174	43-175	43-176
	43-177	43-199	47-26#											
EMS31	43-21	43-146	43-207	43-208	47-27#									
EMS32	43-22	47-28#												
EMS33	43-23	43-219	47-29#											
EMS34	43-24	43-225	43-226	43-227	47-30#									
EMS35	43-25	43-216	47-31#											
EMS36	43-26	43-213	43-214	43-215	47-32#									
EMS37	43-27	47-33#												

EMS4 43-5 43-75 47-6#

M 7

SEQ 0296

EMS40	43-28	43-220	47-34#											
EMS41	43-29	43-190	43-191	47-35#										
EMS42	43-30	43-206	47-36#											
EMS43	43-31	43-209	47-37#											
EMS44	43-32	43-211	43-230	47-38#										
EMS45	43-33	43-210	43-229	47-39#										
EMS46	43-43	43-58	43-120	43-145	43-177	43-185	43-199	43-201	47-40#					
EMS47	43-37	43-44	43-45	43-52	43-60	43-61	43-66	43-117	43-118	43-147	43-186	43-187	43-188	43-192
	43-193	43-222	47-41#											
EMS5	43-6	43-7	47-7#											
EMS50	43-46	43-48	43-64	43-70	43-132	43-135	43-139	43-141	43-181	43-185	43-189	43-200	43-201	43-221
	47-42#													
EMS51	43-48	43-55	43-70	43-130	43-139	43-142	43-179	43-189	43-221	47-43#				
EMS52	43-55	43-142	43-143	43-148	43-149	43-178	43-179	43-185	43-201	43-228	43-229	43-230	43-231	43-233
	47-44#													
EMS53	43-37	43-38	43-39	43-40	43-44	43-45	43-46	43-47	43-62	43-64	43-65	43-73	43-140	43-141
	43-181	43-182	43-184	43-186	43-187	43-190	43-191	43-192	43-193	43-194	43-195	43-200	43-202	43-203
	43-204	43-206	43-207	43-216	43-218	43-219	43-220	43-226	47-45#					
EMS54	43-226	47-46#												
EMS55	43-227	47-47#												
EMS56	43-50	43-51	43-54	43-62	43-63	43-67	43-119	43-143	43-144	43-178	43-182	43-183	43-194	43-195
	43-223	43-228	43-234	47-48#										
EMS57	43-190	43-191	43-192	43-193	43-194	43-195	43-196	43-197	43-198	43-199	43-200	43-201	43-202	43-203
	43-204	43-205	43-206	43-207	43-208	43-209	43-210	43-211	43-213	43-214	43-215	43-220	43-221	43-222
	43-223	43-228	43-229	43-230	47-49#									
EMS6	43-6	47-8#												
EMS60	43-224	47-50#												
EMS61	43-224	47-51#												
EMS62	43-224	47-52#												
EMS63	43-42	43-149	43-190	47-53#										
EMS64	43-8	43-14	43-41	43-144	43-145	43-146	43-147	43-199	47-54#					
EMS65	43-41	47-55#												
EMS66	43-34	43-218	47-56#											
EMS67	43-43	43-44	43-45	43-46	43-47	43-48	43-49	43-50	43-51	43-52	43-54	47-57#		
EMS7	43-10	47-9#												
EMS70	43-48	43-50	43-52	43-54	43-66	43-70	43-73	43-107	43-110	43-139	43-182	43-189	43-194	43-205
	43-221	47-58#												
EMS71	43-51	43-59	43-63	43-183	43-195	43-208	47-59#							
EMS72	43-36	43-48	43-51	43-55	43-56	43-57	43-60	43-61	43-61	43-63	43-108	43-109	43-134	43-135
	43-139	43-140	43-142	43-179	43-183	43-187	43-193	43-195	43-232	47-60#				
EMS73	43-49	43-54	43-67	43-68	43-136	43-180	43-188	43-222	43-223	43-234	43-235	47-61#		
EMS74	43-69	43-70	43-134	43-189	43-217	43-221	47-62#							
EMS75	43-35	43-66	43-67	43-137	43-138	43-188	43-222	43-223	43-234	47-63#				
EMS76	43-35	43-36	43-37	43-38	43-39	43-40	47-64#							
EMS77	43-38	43-72	43-148	43-198	47-65#									
EMT1	7-4	43-3#												
EMT10	7-25	43-10#												
EMT100	7-198	43-66#												
EMT101	7-201	43-67#												
EMT102	7-204	43-68#												
EMT103	7-207	43-69#												
EMT104	7-210	43-70#												
EMT105	7-213	43-71#												
EMT106	7-216	43-72#												
EMT107	7-219	43-73#												
EMT11	7-28	43-11#												

EMT111 7-225 43-75#
EMT112 7-228 43-76#
EMT113 7-231 43-77#
EMT114 7-234 43-78#
EMT115 7-237 43-79#
EMT116 7-240 43-80#
EMT117 7-243 43-81#
EMT12 7-31 43-12#
EMT120 7-246 43-82#
EMT121 7-249 43-83#
EMT122 7-252 43-84#
EMT123 7-255 43-85#
EMT124 7-258 43-86#
EMT125 7-261 43-87#
EMT126 7-264 43-88#
EMT127 7-267 43-89#
EMT13 7-34 43-13#
EMT130 7-270 43-90#
EMT131 7-273 43-91#
EMT132 7-276 43-92#
EMT133 7-279 43-93#
EMT134 7-282 43-94#
EMT135 7-285 43-95#
EMT136 7-288 43-96#
EMT137 7-291 43-97#
EMT14 7-37 43-14#
EMT140 7-294 43-98#
EMT141 7-297 43-99#
EMT142 7-300 43-100#
EMT143 7-303 43-101#
EMT144 7-306 43-102#
EMT145 7-309 43-103#
EMT146 7-312 43-104#
EMT147 7-315 43-105#
EMT15 7-40 43-15#
EMT150 7-318 43-106#
EMT151 7-321 43-107#
EMT152 7-324 43-108#
EMT153 7-327 43-109#
EMT154 7-330 43-110#
EMT155 7-333 43-111#
EMT156 7-336 43-112#
EMT157 7-339 43-113#
EMT16 7-43 43-16#
EMT160 7-342 43-114#
EMT161 7-345 43-115#
EMT162 7-348 43-116#
EMT163 43-117#
EMT164 7-355 43-118#
EMT165 7-358 43-119#
EMT166 7-361 43-120#
EMT167 7-364 43-121#
EMT17 7-46 43-17#
EMT170 43-122#
EMT171 7-370 43-123#
EMT172 7-373 43-124#

EMT174 7-379 43-126#
EMT175 7-382 43-127#
EMT176 7-385 43-128#
EMT177 7-388 43-129#
EMT2 7-7 43-4#
EMT20 7-49 43-18#
EMT200 7-391 43-130#
EMT201 7-394 43-131#
EMT202 7-397 43-132#
EMT203 7-400 43-133#
EMT204 7-403 43-134#
EMT205 7-406 43-135#
EMT206 7-409 43-136#
EMT207 7-412 43-137#
EMT21 7-52 43-19#
EMT210 7-415 43-138#
EMT211 7-418 43-139#
EMT212 7-421 43-140#
EMT213 7-424 43-141#
EMT214 7-427 43-142#
EMT215 7-430 43-143#
EMT216 7-433 43-144#
EMT217 7-436 43-145#
EMT22 7-55 43-20#
EMT220 7-439 43-146#
EMT221 7-442 43-147#
EMT222 7-445 43-148#
EMT223 7-448 43-149#
EMT224 43-150#
EMT225 43-151#
EMT226 43-152#
EMT227 43-153#
EMT23 7-58 43-21#
EMT230 43-154#
EMT231 43-155#
EMT232 43-156#
EMT233 43-157#
EMT234 43-158#
EMT235 43-159#
EMT236 43-160#
EMT237 43-161#
EMT24 7-61 43-22#
EMT240 43-162#
EMT241 43-163#
EMT242 43-164#
EMT243 43-165#
EMT244 43-166#
EMT245 43-167#
EMT246 7-505 43-168#
EMT247 7-508 43-169#
EMT25 7-64 43-23#
EMT250 7-511 43-170#
EMT251 7-514 43-171#
EMT252 7-517 43-172#
EMT253 7-520 43-173#
EMT254 7-523 43-174#

EMT255 7-526 43-175#

F 8

SEQ 0302

ERR 4-560# 25-84 25-86 25-102 25-136

J 8

SEQ 0306

12-<17 12-<17# 12-<17# 12-<25 12-<25# 12-<25# 12-<39 L 12-<39# B 12-<39# 12-<46 12-<46# 12-<46# 12-<50 12-<50#

SEQ 0308

35-378 35-381 35-391 35-394 40-77 40-78 40-81 N 8 40-85 40-86

SEQ 0310

LST 4-642# 31-82 34-70

SEQ 0312

RMBAEO 6-0#

G 9

SEQ 0316

CCEEEGGGMMN NPP PRRSSSSSS

RMDTI 6-0# 12-67* 12-69 12-71 12-74 12-76 12-79 I 9 12-81 49-5 49-17

SEQ 0318

12-f66* 16-59* 16-302 17-22 18-73 26-24 26-45[K 9]

SEQ 0320

SW02 4-491 4-491#

M 9

SEQ 0322

12-=96 12-A06 12-A32 12-B13 12-B35 12-C12 12-C26 D 10 12-D04 12-D26 12-E35 12-E40 12-E67 12-E89 12-F68

SEQ 0326

	12-F73	35-342											
WLE	4-577#	4-590	25-221	25-225	25-238	25-253	25-268	35-279	35-311	35-315	35-323	40-81	40-82
WRL	4-563#	25-223	35-318										
XSIZ	9-67#	10-123	10-153										
XXDP	6-0#	9-30*	9-33*	9-34	9-36*	9-41	9-52	9-77	9-79				
ZEROS	12-104	12-205	12-303	12-396	12-834	12-930	12-:25	12-;18	12-C16	12-D08	12-E71	42-38#	

15-84 15-107 .5-139 15-161 15-194 16-211 16-291 H 10 16-295 24-2 25-26 25-34 25-119 26-1 26-3

	31-2	38-1	38-2	38-3	38-4	38-5	38-6	38-7	38-8	38-8	38-8	38-8	38-8	38-9
	38-10	38-11	38-11	38-12										
SWRSU	4-491#	9-19	9-19#											
TAGS	4-111#	5-0												
TRMTRP	38-10#													
TYPBIN	4-491#													
TYPDEC	4-491#	13-20	13-20											
TYPNAM	4-479#	4-491#	9-25											
TYPNUM	4-491#													
TYPOCS	4-491#	9-99	14-23	14-27	14-32	14-34								
TYPOCT	4-491#	10-56	38-8											
TYPTXT	4-491#	9-6	9-39	9-50	9-56	9-57	11-24	13-20	13-20	37-18				
XPER	4-20#	7-4	7-7	7-10	7-13	7-16	7-19	7-22	7-25	7-28	7-31	7-34	7-37	7-40
	7-43	7-46	7-49	7-52	7-55	7-58	7-61	7-64	7-67	7-70	7-73	7-76	7-79	7-82
	7-85	7-88	7-91	7-94	7-97	7-100	7-103	7-106	7-109	7-112	7-115	7-118	7-121	7-124
	7-127	7-131	7-134	7-137	7-140	7-143	7-147	7-151	7-155	7-158	7-161	7-164	7-167	7-170
	7-173	7-176	7-179	7-183	7-186	7-189	7-192	7-195	7-198	7-201	7-204	7-207	7-210	7-213
	7-216	7-219	7-222	7-225	7-228	7-231	7-234	7-237	7-240	7-243	7-246	7-249	7-252	7-255
	7-258	7-261	7-264	7-267	7-270	7-273	7-276	7-279	7-282	7-285	7-288	7-291	7-294	7-297
	7-300	7-303	7-306	7-309	7-312	7-315	7-318	7-321	7-324	7-327	7-330	7-333	7-336	7-339
	7-342	7-345	7-348	7-352	7-355	7-358	7-361	7-364	7-367	7-370	7-373	7-376	7-379	7-382
	7-385	7-388	7-391	7-394	7-397	7-400	7-403	7-406	7-409	7-412	7-415	7-418	7-421	7-424
	7-427	7-430	7-433	7-436	7-439	7-442	7-445	7-448	7-451	7-454	7-457	7-460	7-463	7-466
	7-469	7-472	7-475	7-478	7-481	7-484	7-487	7-490	7-493	7-496	7-499	7-502	7-505	7-508
	7-511	7-514	7-517	7-520	7-523	7-526	7-529	7-532	7-535	7-538	7-541	7-544	7-547	7-550
	7-553	7-557	7-560	7-564	7-567	7-570	7-574	7-578	7-582	7-587	7-591	7-595	7-598	7-601
	7-604	7-607	7-610	7-613	7-616	7-619	7-622	7-625	7-628	7-631	7-634	7-637	7-640	7-643
	7-646	7-649	7-652	7-655	7-658	7-661	7-664	7-667	7-670	7-673	7-676	7-679	7-682	7-685
	7-688	7-691	7-694	7-697	7-700	7-703	7-706	7-709	7-712	7-715	7-718	7-721	7-724	