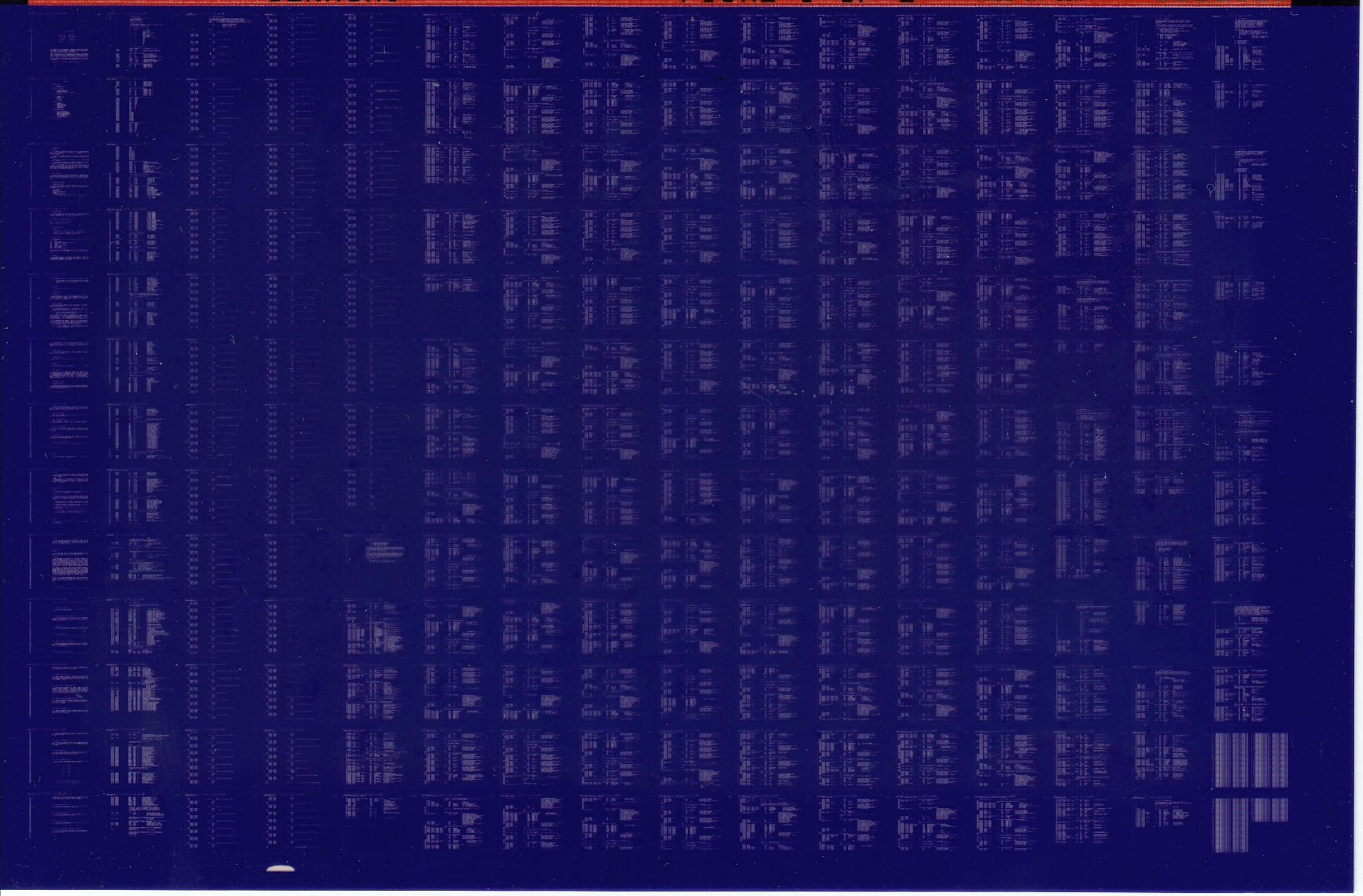
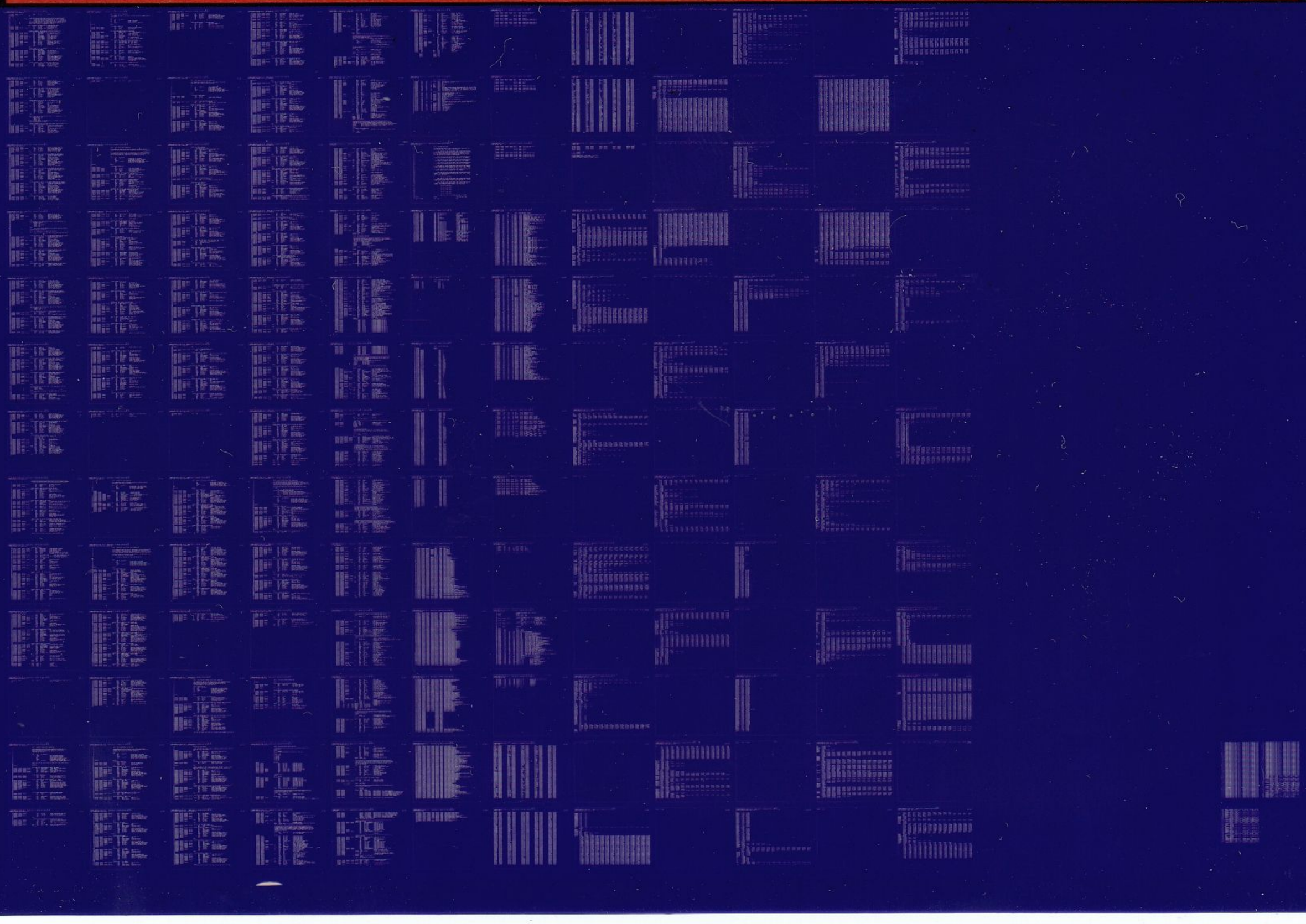
RM05/3/2

RM05/3/2 FCTNL TST 3 AH-F928A-MC
CZRM0A0
FICHE 1 OF 2



RM05/3/2

RM05/3/2 FCTNL TST 3 AH-F928A-MC
CZRM0A0
FICHE 2 OF 2



```
.REM    \
```

IDENTIFICATION

PRODUCT CODE: AC-F927A-MC

PRODUCT NAME: CZRMOA0 RM05/3/2 FCTNL TST 3

DATE CREATED: APRIL 1980

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

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 & 2

3.0 OPERATING PROCEDURE

3.1 LOADING

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

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

3.2 SWITCH OPTIONS

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

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

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

3.3 STARTING

THE PROGRAM 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 DATA TESTS

PURPOSE:

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

PROCEDURE:

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

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

NOTE:
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.

WRITE, READ ZEROS TEST

THE TEST WRITES AND READS AN ALL ZEROS DATA FIELD, CAUSING

THE DRIVE TO USE NORMAL WRITE GATE THOUGHOUT THE WRITE PROCESS.

WRITE, WRITE CHECK ZEROS TEST

THE TEST WRITES AND WRITE CHECKS AN ALL ZEROS DATA FIELD, VERIFYING THAT THERE ARE NO ERRORS.

WRITE, WRITE CHECK ZEROS W/ WCE ERROR TEST

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

WRITE, READ ONES TEST

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

WRITE, WRITE CHECK ONES TEST

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

WRITE, WRITE CHECK ONES W/ WCE ERROR TEST

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

WRITE, WRITE CHECK MULTIPLE SECTORS TEST

THE TEST SEEKS TO CYLINDER 0 TO INSURE THERE IS NO HEAD

MOTION DURING DATA TRANSFER. THE WRITE DATA COMMAND FOLLOWS, WITH THE WORD COUNT EQUAL TO MULTIPLE SECTORS. THE SAME SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND.

WRITE, READ WITH IMPLIED SEEK TEST

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

WRITE, WRITE CHECK WITH HEAD SWITCHING TEST

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

WRITE, WRITE CHECK WITH MID-TRANSFER SEEK TEST

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

WRITE, READ W/ HCE ERROR TEST

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

WRITE, READ W/ HCI TEST

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

WRITE, READ W/ IVC ERROR TEST

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

WRITE, READ W/ ABORT TEST

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

WRITE, READ EACH CURRENT LEVEL TEST

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

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

WRITE, WRITE CHECK W/ CURRENT LEVEL SWITCHING TEST

THE TEST WRITES 2 SECTORS STARTING WITH THE LAST SECTOR OF

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

WRITE, READ EARLY PEAK SHIFT TEST

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

WRITE, READ MIXED PEAK SHIFT TEST

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

WRITE, READ EACH TRACK TEST

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

READ, WRITES CHECK MULTIPLE SECTORS IN OFFSET MODE TEST

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

```
     1                          ;PROGRAM REVISION #001
   486
   487                          .TITLE  CZRMOAO RM05/3/2 FCTNL TST 3
                                ;*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
                        TA16    = BIT12                 ;TRACK ADDRESS 16.
                        TA8     = BIT11                 ;TRACK ADDRESS 8.
                        TA4     = BIT10                 ;TRACK ADDRESS 4
                        TA2     = BIT09                 ;TRACK ADDRESS 2
                        TA1     = BIT08                 ;TRACK ADDRESS 1

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

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

                        ;*RMDS  DRIVE STATUS REGISTER

                        ATA     = BIT15                 ;ATTENTION ACTIVE
                        ERR     = BIT14                 ;COMPOSITE ERROR
                        PIP     = BIT13                 ;POSITIONING IN PROGRESS
                        MOL     = BIT12                 ;MEDIUM ON LINE
                        WRL     = BIT11                 ;WRITE LOCK
                        LBT     = BIT10                 ;LAST BLOCK TRANSFERRED
                        PGM     = BIT09                 ;PROGRAMMABLE
                        DPR     = BIT08                 ;DRIVE PRESENT
                        DRY     = BIT07                 ;DRIVE READY
                        VV      = BIT06                 ;VOLUME VALID
                        OM      = BIT00                 ;OFFSET MODE ACTIVE

                        ;*RMER1 ERROR REGISTER #1
```

```
;72
573          100000          DCK     = BIT15                 ;DATA CHECK ERROR
574          040000          UNS     = BIT14                 ;DRIVE UNSAFE
575          020000          OPI     = BIT13                 ;OPERATION INCOMPLETE
576          010000          DTE     = BIT12                 ;DRIVE TIMING ERROR
577          004000          WLE     = BIT11                 ;WRITE LOCK ERROR
578          002000          IAE     = BIT10                 ;INVALID ADDRESS ERROR
579          001000          AOE     = BIT09                 ;ADDRESS OVERFLOW ERROR
580          000400          HCRC    = BIT08                 ;HEADER CRC ERROR
581          000200          HCE     = BIT07                 ;HEADER COMPARE ERROR
582          000100          ECH     = BIT06                 ;ECC 'HARD' ERROR
583          000040          WCF     = BIT05                 ;WRITE CLOCK FAILURE
584          000020          FER     = BIT04                 ;FORMAT ERROR
585          000010          PAR     = BIT03                 ;PARITY ERROR
586          000004          RMR     = BIT02                 ;REGISTER MODIFICATION REFUSED
587          000002          ILR     = BIT01                 ;ILLEGAL REGISTER
588          000001          ILF     = BIT00                 ;ILLEGAL FUNCTION
589
590          115760          NDTMSK  = DCK!DTE!WLE!AOE!HCRC!HCE!ECH!WCF!FER
591                          ;'NDTMSK' IS USED TO MASK ERROR REGISTER 1 DURING NON - DATA
592                          ;COMMANDS, I.E., HOUSEKEEPING AND POSITIONING COMMANDS
593
594                          ;*RMAS  ATTENTION SUMMARY REGISTER
595
596          000377          ATNMSK  = 377                   ;MASK FOR ATTENTION BITS
597
598                          ;*RMLA  LOOK AHEAD REGISTER
599
600          002000          SC4     = BIT10                 ;SECTOR COUNT = 16
601          001000          SC3     = BIT09                 ;SECTOR COUNT = 8
602          000400          SC2     = BIT08                 ;SECTOR COUNT = 4
603          000200          SC1     = BIT07                 ;SECTOR COUNT = 2
604          000100          SC0     = BIT06                 ;SECTOR COUNT = 1
605
606          003700          SCTMSK  = 003700                ;SECTOR COUNT MASK
607
608                          ;*RMMR1 MAINTENANCE REGISTER #1
609
610                          ;WRITE ONLY BITS
611          100000          DBCK    = BIT15                 ;DEBUG CLOCK
612          040000          DBEN    = BIT14                 ;DEBUG CLOCK ENABLE
613          020000          DEBL    = BIT13                 ;DIAGNOSTIC END OF BLOCK
614          010000          DTO     = BIT12                 ;DIAGNOSTIC TIMEOUT
615          004000          MCLK    = BIT11                 ;MAINTENANCE CLOCK
616          002000          MRD     = BIT10                 ;READ DATA
617          001000          MUR     = BIT09                 ;UNIT READY
618          000400          MOC     = BIT08                 ;ON CYLINDER
619          000200          MSER    = BIT07                 ;SEEK ERROR
620          000100          MDF     = BIT06                 ;DRIVE FAULT
621          000040          MS      = BIT05                 ;SECTOR PULSE
622          000010          MWP     = BIT03                 ;WRITE PROTECT
623          000004          MI      = BIT02                 ;INDEX PULSE
624          000002          MSC     = BIT01                 ;SECTOR COMPARE
625          000001          DMD     = BIT00                 ;DIAGNOSTIC MODE
626
627                          ;READ ONLY BITS
628          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
```

```
                        ;*RMCS2 RH CONTROL STATUS REGISTER #2

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

                        ;UNIT SELECT MASK

000007                  UNTMSK  = 7             ;UNIT SELECT MASK

                        ;*RMCS3 RH70 CONTROL STATUS REGISTER #3

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

                        .SBTTL  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     037650                          $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            $SCNT:  .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            RMCS10: .WORD   0       ;CONTROL, STATUS REGISTER #1
      001412  000000            RMWC0:  .WORD   0       ;WORD COUNT REGISTER
      001414  000000            RMBA0:  .WORD   0       ;BUS ADDRESS REGISTER
      001416  000000            RMDA0:  .WORD   0       ;DISK ADDRESS REGISTER
      001420  000000            RMCS20: .WORD   0       ;CONTROL, STATUS REGISTER #2
      001422  000000            RMDS0:  .WORD   0       ;DRIVE STATUS REGISTER
      001424  000000            RMER10: .WORD   0       ;ERROR REGISTER #1
      001426  000000            RMAS0:  .WORD   0       ;ATTENTION SUMMARY REGISTER
      001430  000000            RMLA0:  .WORD   0       ;LOOK AHEAD REGISTER
      001432  000000            RMDB0:  .WORD   0       ;DATA BUFFER
      001434  000000            RMMR10: .WORD   0       ;MAINTENANCE REGISTER #1
      001436  000000            RMDT0:  .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  072254                          EMT1
        001602  076346                          EHT1
        001604  076472                          EDT1
        001606  076562                          EFT1

    5
    6                                   ;ERROR  2       DEVICE WENT UNAVAILABLE
    7
        001610  072260                          EMT2
        001612  076346                          EHT1
        001614  076472                          EDT1
        001616  076562                          EFT1

    8
    9                                   ;ERROR  3       DEVICE WENT NONEXISTENT
   10
        001620  072266                          EMT3
        001622  076346                          EHT1
        001624  076472                          EDT1
        001626  076562                          EFT1

   11
   12                                   ;ERROR  4       CONTROLLER NOT READY
   13
        001630  072274                          EMT4
        001632  076346                          EHT1
        001634  076472                          EDT1
        001636  076562                          EFT1

   14
   15                                   ;ERROR  5       DRIVE NOT READY AND GO NOT RESET
   16
        001640  072302                          EMT5
        001642  076346                          EHT1
        001644  076472                          EDT1
        001646  076562                          EFT1

   17
```

```
                                ;ERROR  6       UNEXPECTED VALUE FOR ''ATA'' STATUS
001650  072310                          EMT6
001652  076346                          EHT1
001654  076472                          EDT1
001656  076562                          EFT1

                                ;ERROR  7       BUS TIMEOUT TRYING TO READ OR WRITE REGISTER
001660  072316                          EMT7
001662  000000                          0
001664  000000                          0
001666  000000                          0

                                ;ERROR  10      DRIVE NOT READY BUT GO IS RESET
001670  072324                          EMT10
001672  076346                          EHT1
001674  076472                          EDT1
001676  076562                          EFT1

                                ;ERROR  11      GO NOT RESET BUT DRIVE IS READY
001700  072330                          EMT11
001702  076346                          EHT1
001704  076472                          EDT1
001706  076562                          EFT1

                                ;ERROR  12      INCORRECT FUNCTION CODE
001710  072334                          EMT12
001712  076346                          EHT1
001714  076472                          EDT1
001716  076562                          EFT1

                                ;ERROR  13      PARITY ERROR READING REMOTE REGISTERS
001720  072342                          EMT13
001722  076346                          EHT1
001724  076472                          EDT1
001726  076562                          EFT1

                                ;ERROR  14      TRANSFER ERROR IS INCORRECT
001730  072354                          EMT14
001732  076346                          EHT1
001734  076472                          EDT1
001736  076562                          EFT1

                                ;ERROR  15      INCORRECT WORD COUNT
```

```
 40
    001740  072362                          EMT15
    001742  076346                          EHT1
    001744  076472                          EDT1
    001746  076562                          EFT1

 41
 42                                 ;ERROR  16      INCORRECT BUS ADDRESS
 43
    001750  072370                          EMT16
    001752  076346                          EHT1
    001754  076472                          EDT1
    001756  076562                          EFT1

 44
 45                                 ;ERROR  17      INCORRECT LBT STATUS
 46
    001760  072400                          EMT17
    001762  076346                          EHT1
    001764  076472                          EDT1
    001766  076562                          EFT1

 47
 48                                 ;ERROR  20      INCORRECT AQE
 49
    001770  072410                          EMT20
    001772  076346                          EHT1
    001774  076472                          EDT1
    001776  076562                          EFT1

 50
 51                                 ;ERROR  21      INCORRECT DISK ADDRESS
 52
    002000  072420                          EMT21
    002002  076346                          EHT1
    002004  076472                          EDT1
    002006  076562                          EFT1

 53
 54                                 ;ERROR  22      INCORRECT CYLINDER ADDRESS
 55
    002010  072430                          EMT22
    002012  076346                          EHT1
    002014  076472                          EDT1
    002016  076562                          EFT1

 56
 57                                 ;ERROR  23      INCORRECT WLE STATUS
 58
    002020  072440                          EMT23
    002022  076346                          EHT1
    002024  076472                          EDT1
    002026  076562                          EFT1

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

```
002030  072450                          EMT24
002032  076346                          EHT1
002034  076472                          EDT1
002036  076562                          EFT1

62
63                              ;ERROR  25      INCORRECT WCF STATUS
64
002040  072460                          EMT25
002042  076346                          EHT1
002044  076472                          EDT1
002046  076562                          EFT1

65
66                              ;ERROR  26      INCORRECT WCE STATUS
67
002050  072470                          EMT26
002052  076346                          EHT1
002054  076472                          EDT1
002056  076562                          EFT1

68
69                              ;ERROR  27      INCORRECT MDPE STATUS
70
002060  072500                          EMT27
002062  076346                          EHT1
002064  076472                          EDT1
002066  076562                          EFT1

71
72                              ;ERROR  30      INCORRECT DCK STATUS
73
002070  072510                          EMT30
002072  076346                          EHT1
002074  076472                          EDT1
002076  076562                          EFT1

74
75                              ;ERROR  31      INCORRECT ECH STATUS
76
002100  072520                          EMT31
002102  076346                          EHT1
002104  076472                          EDT1
002106  076562                          EFT1

77
78                              ;ERROR  32      DLT SHOULD NOT BE SET
79
002110  072530                          EMT32
002112  076346                          EHT1
002114  076472                          EDT1
002116  076562                          EFT1

80
81                              ;ERROR  33      MXF SHOULD NOT BE SET
82
002120  072540                          EMT33
```

```
        002122  076346                          EHT1
        002124  076472                          EDT1
        002126  076562                          EFT1

     83
     84                                 ;ERROR  34      DTE SHOULD NOT BE SET
     85
        002130  072550                          EMT34
        002132  076346                          EHT1
        002134  076472                          EDT1
        002136  076562                          EFT1

     86
     87                                 ;ERROR  35      INCORRECT HCRC STATUS
     88
        002140  072560                          EMT35
        002142  076346                          EHT1
        002144  076472                          EDT1
        002146  076562                          EFT1

     89
     90                                 ;ERROR  36      INCORRECT HCE STATUS
     91
        002150  072570                          EMT36
        002152  076346                          EHT1
        002154  076472                          EDT1
        002156  076562                          EFT1

     92
     93                                 ;ERROR  37      INCORRECT FER STATUS
     94
        002160  072600                          EMT37
        002162  076346                          EHT1
        002164  076472                          EDT1
        002166  076562                          EFT1

     95
     96                                 ;ERROR  40      DPE SHOULD NOT BE SET (NOT A DATA COMMAND)
     97
        002170  072610                          EMT40
        002172  076346                          EHT1
        002174  076472                          EDT1
        002176  076562                          EFT1

     98
     99                                 ;ERROR  41      LOST 'MOL' DURING PACK ACKNOWLEDGE
    100
        002200  072616                          EMT41
        002202  076346                          EHT1
        002204  076472                          EDT1
        002206  076562                          EFT1

    101
    102                                 ;ERROR  42      UNSAFE ERROR DURING PACK ACKNOWLEDGE
    103
        002210  072626                          EMT42
        002212  076346                          EHT1
```

```
        002214  076472                          EDT1
        002216  076562                          EFT1

    104
    105                             ;ERROR  43      ''DPI'' ERROR DURING PACK ACKNOWLEDGE
    106
        002220  072640                          EMT43
        002222  076346                          EHT1
        002224  076472                          EDT1
        002226  076562                          EFT1

    107
    108                             ;ERROR  44      ''RMR'' ERROR DURING PACK ACKNOWLEDGE
    109
        002230  072650                          EMT44
        002232  076346                          EHT1
        002234  076472                          EDT1
        002236  076562                          EFT1

    110
    111                             ;ERROR  45      ''ILR'' ERROR DURING PACK ACKNOWLEDGE
    112
        002240  072660                          EMT45
        002242  076346                          EHT1
        002244  076472                          EDT1
        002246  076562                          EFT1

    113
    114                             ;ERROR  46      ''ILF'' ERROR DURING PACK ACKNOWLEDGE
    115
        002250  072670                          EMT46
        002252  076346                          EHT1
        002254  076472                          EDT1
        002256  076562                          EFT1

    116
    117                             ;ERROR  47      COMPOSITE ERROR STATUS IS INCORRECT
    118
        002260  072700                          EMT47
        002262  076346                          EHT1
        002264  076472                          EDT1
        002266  076562                          EFT1

    119
    120                             ;ERROR  50      PARITY ERROR WRITING REMOTE REGISTERS
    121
        002270  072706                          EMT50
        002272  076346                          EHT1
        002274  076472                          EDT1
        002276  076562                          EFT1

    122
    123                             ;ERROR  51      INCORRECT IAE STATUS DURING SEEK COMMAND
    124
        002300  072716                          EMT51
        002302  076346                          EHT1
        002304  076472                          EDT1
```

```
        002306  076562                          EFT1

    125
    126                             ;ERROR  52      OPI ERROR DURING SEEK - MEDIUM IS NOT ON LINE
    127
        002310  072730                          EMT52
        002312  076346                          EHT1
        002314  076472                          EDT1
        002316  076562                          EFT1

    128
    129                             ;ERROR  53      OPI ERROR DURING SEEK - MEDIUM IS ON LINE, ASSUME
    130                             ;               ON CYLINDER LATCH DIDN'T RESET
    131
        002320  072746                          EMT53
        002322  076346                          EHT1
        002324  076472                          EDT1
        002326  076562                          EFT1

    132
    133                             ;ERROR  54      SEEK INCOMPLETE ERROR DURING SEEK COMMAND
    134
        002330  072764                          EMT54
        002332  076346                          EHT1
        002334  076472                          EDT1
        002336  076562                          EFT1

    135
    136                             ;ERROR  55      DEVICE CHECK DURING SEEK COMMAND
    137
        002340  072774                          EMT55
        002342  076346                          EHT1
        002344  076472                          EDT1
        002346  076562                          EFT1

    138
    139                             ;ERROR  56      PIP IS STILL SET AFTER SEEK - SKI IS RESET
    140
        002350  073006                          EMT56
        002352  076346                          EHT1
        002354  076472                          EDT1
        002356  076562                          EFT1

    141
    142                             ;ERROR  57      ATA DID NOT SET DURING SEEK COMMAND
    143
        002360  073024                          EMT57
        002362  076346                          EHT1
        002364  076472                          EDT1
        002366  076562                          EFT1

    144
    145                             ;ERROR  60      IVC ERROR DURING SEEK COMMAND - LOST
    146                             ;               VOLUME VALID
    147
        002370  073034                          EMT60
        002372  076346                          EHT1
```

```
                002374  076472                          EDT1
                002376  076562                          EFT1

    148
    149                                 ;ERROR  61      ERRONEOUS IVC ERROR DURING SEEK COMMAND -
    150                                 ;               VOLUME VALID IS STIL SET
    151
                002400  073052                          EMT61
                002402  076346                          EHT1
                002404  076472                          EDT1
                002406  076562                          EFT1

    152
    153                                 ;ERROR  62      MOL IS ZERO, BUT OPI WAS NOT
    154                                 ;               REPORTED DURING SEEK COMMAND
    155
                002410  073072                          EMT62
                002412  076346                          EHT1
                002414  076472                          EDT1
                002416  076562                          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  073110                          EMT64
                002432  076346                          EHT1
                002434  076472                          EDT1
                002436  076562                          EFT1

    162
    163                                 ;ERROR  65      DRIVE EXECUTED A SEEK WITH ERROR SET
    164
                002440  073130                          EMT65
                002442  076346                          EHT1
                002444  076472                          EDT1
                002446  076562                          EFT1

    165
    166                                 ;ERROR  66      UNEXPECTED ERROR SET IN RMER1
    167
                002450  073150                          EMT66
                002452  076346                          EHT1
                002454  076472                          EDT1
                002456  076562                          EFT1

    168
    169                                 ;ERROR  67      UNEXPECTED ERROR SET IN RMER2
    170
                002460  073162                          EMT67
```

```
002462  076346                          EHT1
002464  076472                          EDT1
002466  076562                          EFT1

171
172                             ;ERROR  70      ERRONEOUS ''IAE'' ERROR DURING RECALIBRATE
173
002470  073174                          EMT70
002472  076346                          EHT1
002474  076472                          EDT1
002476  076562                          EFT1

174
175                             ;ERROR  71      ''ILF'' ERROR DURING RECALIBRATE
176
002500  073204                          EMT71
002502  076346                          EHT1
002504  076472                          EDT1
002506  076562                          EFT1

177
178                             ;ERROR  72      'OPI'' ERROR DURING RECALIBRATE DUE TO 'MOL'' = 0
179
002510  073214                          EMT72
002512  076346                          EHT1
002514  076472                          EDT1
002516  076562                          EFT1

180
181                             ;ERROR  73      'OPI'' ERROR DURING RECALIBRATE BECAUSE ON
182                             ;               CYLINDER DIDNT DROP
183
002520  073232                          EMT73
002522  076346                          EHT1
002524  076472                          EDT1
002526  076562                          EFT1

184
185                             ;ERROR  74      ''IVC'' ERROR DURING RECALIBRATE - 'VV'' = 0
186
002530  073250                          EMT74
002532  076346                          EHT1
002534  076472                          EDT1
002536  076562                          EFT1

187
188                             ;ERROR  75      ERRONEOUS ''IVC'' ERROR DURING RECALIBRATE - 'VV'' = 1
189
002540  073260                          EMT75
002542  076346                          EHT1
002544  076472                          EDT1
002546  076562                          EFT1

190
191                             ;ERROR  76      'SKI'' ERROR DURING RECALIBRATE
192
002550  073300                          EMT76
```

```
        002552  076346                          EHT1
        002554  076472                          EDT1
        002556  076562                          EFT1

    193
    194                             ;ERROR  77      'DVC'' OCCURRED DURING RECALIBRATE
    195
        002560  073310                          EMT77
        002562  076346                          EHT1
        002564  076472                          EDT1
        002566  076562                          EFT1

    196
    197                             ;ERROR  100     LOST 'MOL'' DURING RECALIBRATE - 'DPI'' - 0
    198
        002570  073322                          EMT100
        002572  076346                          EHT1
        002574  076472                          EDT1
        002576  076562                          EFT1

    199
    200                             ;ERROR  101     LOST 'VV'' DURING RECALIBRATE - ''IVC'' = 0
    201
        002600  073340                          EMT101
        002602  076346                          EHT1
        002604  076472                          EDT1
        002606  076562                          EFT1

    202
    203                             ;ERROR  102     ''ATA'' DID NOT SET DURING RECALIBRATE
    204
        002610  073356                          EMT102
        002612  076346                          EHT1
        002614  076472                          EDT1
        002616  076562                          EFT1

    205
    206                             ;ERROR  103     'DM'' DID NOT RESET DURING RECALIBRATE
    207
        002620  073366                          EMT103
        002622  076346                          EHT1
        002624  076472                          EDT1
        002626  076562                          EFT1

    208
    209                             ;ERROR  104     'PIP'' IS STIL SET AFTER RECALIBRATE
    210
        002630  073400                          EMT104
        002632  076346                          EHT1
        002634  076472                          EDT1
        002636  076562                          EFT1

    211
    212                             ;ERROR  105     UNEXPECTED ''ILR'' ERROR DURING RECALIBRATE
    213
        002640  073416                          EMT105
        002642  076346                          EHT1
```

```
        002644  076472                          EDT1
        002646  076562                          EFT1

    214
    215                             ;ERROR  106     UNEXPECTED 'RMR' ERROR DURING RECALIBRATE
    216
        002650  073426                          EMT106
        002652  076346                          EHT1
        002654  076472                          EDT1
        002656  076562                          EFT1

    217
    218                             ;ERROR  107     'UNS' ERROR DURING RECALIBRATE - AC POWER IS LOW
    219
        002660  073436                          EMT107
        002662  076346                          EHT1
        002664  076472                          EDT1
        002666  076562                          EFT1

    220
    221                             ;ERROR  110     CANNOT ACCESS MASSBUS CONTROLLER VIA UNIBUS
    222
        002670  073456                          EMT110
        002672  076370                          EHT110
        002674  076510                          EDT110
        002676  076600                          EFT110

    223
    224                             ;ERROR  111     NONEXISTENT DEVICE
    225
        002700  073470                          EMT111
        002702  076374                          EHT111
        002704  076512                          EDT111
        002706  076602                          EFT111

    226
    227                             ;ERROR  112     DEVICE NOT AVAILABLE
    228
        002710  073476                          EMT112
        002712  076374                          EHT111
        002714  076512                          EDT111
        002716  076602                          EFT111

    229
    230                             ;ERROR  113     BUS TIMEOUT-NED STATUS FAILURE
    231
        002720  073504                          EMT113
        002722  000000                          0
        002724  000000                          0
        002726  000000                          0

    232
    233                             ;ERROR  114     DEVICE NOT AN RM05/3/2
    234
        002730  073520                          EMT114
        002732  076400                          EHT114
        002734  076514                          EDT114
```

```
        002736  076604                          EFT114

    235
    236                             ;ERROR  115     RMCS1 NOT INITIALIZED BY UNIBUS
    237
        002740  073526                          EMT115
        002742  076346                          EHT1
        002744  076472                          EDT1
        002746  076562                          EFT1

    238
    239                             ;ERROR  116     RMBA NOT INITIALIZED BY UNIBUS
    240
        002750  073536                          EMT116
        002752  076346                          EHT1
        002754  076472                          EDT1
        002756  076562                          EFT1

    241
    242                             ;ERROR  117     RMCS2 NOT INITIALIZED BY UNIBUS
    243
        002760  073546                          EMT117
        002762  076346                          EHT1
        002764  076472                          EDT1
        002766  076562                          EFT1

    244
    245                             ;ERROR  120     RMER1 NOT INITIALIZED BY UNIBUS
    246
        002770  073556                          EMT120
        002772  076346                          EHT1
        002774  076472                          EDT1
        002776  076562                          EFT1

    247
    248                             ;ERROR  121     RMAS NOT INITIALIZED BY UNIBUS
    249
        003000  073566                          EMT121
        003002  076346                          EHT1
        003004  076472                          EDT1
        003006  076562                          EFT1

    250
    251                             ;ERROR  122     RMMR1 NOT INITIALIZED BY UNIBUS
    252
        003010  073576                          EMT122
        003012  076346                          EHT1
        003014  076472                          EDT1
        003016  076562                          EFT1

    253
    254                             ;ERROR  123     RMDS NOT INITIALIZED BY UNIBUS
    255
        003020  073606                          EMT123
        003022  076346                          EHT1
        003024  076472                          EDT1
        003026  076562                          EFT1
```

```
                                ;ERROR  124     RMEC2 NOT INITIALIZED BY UNIBUS

003030  073616                          EMT124
003032  076346                          EHT1
003034  076472                          EDT1
003036  076562                          EFT1

                                ;ERROR  125     RMMR2 NOT INITIALIZED BY UNIBUS

003040  073626                          EMT125
003042  076346                          EHT1
003044  076472                          EDT1
003046  076562                          EFT1

                                ;ERROR  126     RMCS1 NOT CLEARED BY CONTROLLER CLEAR

003050  073636                          EMT126
003052  076346                          EHT1
003054  076472                          EDT1
003056  076562                          EFT1

                                ;ERROR  127     RMBA NOT CLEARED BY CONTROLLER CLEAR

003060  073650                          EMT127
003062  076346                          EHT1
003064  076472                          EDT1
003066  076562                          EFT1

                                ;ERROR  130     RMCS2 NOT CLEARED BY CONTROLLER CLEAR

003070  073662                          EMT130
003072  076346                          EHT1
003074  076472                          EDT1
003076  076562                          EFT1

                                ;ERROR  131     RMER1 NOT CLEARED BY CONTROLLER CLEAR

003100  073674                          EMT131
003102  076346                          EHT1
003104  076472                          EDT1
003106  076562                          EFT1

                                ;ERROR  132     RMAS NOT CLEARED BY CONTROLLER CLEAR

003110  073706                          EMT132
003112  076346                          EHT1
003114  076472                          EDT1
003116  076562                          EFT1
```

```
277
278                                     ;ERROR  133     RMMR1 NOT CLEARED BY CONTROLLER CLEAR
279
    003120  073720                              EMT133
    003122  076346                              EHT1
    003124  076472                              EDT1
    003126  076562                              EFT1

280
281                                     ;ERROR  134     RMDS NOT CLEARED BY CONTROLLER CLEAR
282
    003130  073732                              EMT134
    003132  076346                              EHT1
    003134  076472                              EDT1
    003136  076562                              EFT1

283
284                                     ;ERROR  135     RMEC2 NOT CLEARED BY CONTROLLER CLEAR
285
    003140  073744                              EMT135
    003142  076346                              EHT1
    003144  076472                              EDT1
    003146  076562                              EFT1

286
287                                     ;ERROR  136     RMMR2 NOT CLEARED BY CONTROLLER CLEAR
288
    003150  073756                              EMT136
    003152  076346                              EHT1
    003154  076472                              EDT1
    003156  076562                              EFT1

289
290                                     ;ERROR  137     RMCS1 NOT CLEARED BY ERROR CLEAR
291
    003160  073770                              EMT137
    003162  076346                              EHT1
    003164  076472                              EDT1
    003166  076562                              EFT1

292
293                                     ;ERROR  140     RMCS2 NOT CLEARED BY ERROR CLEAR
294
    003170  074000                              EMT140
    003172  076346                              EHT1
    003174  076472                              EDT1
    003176  076562                              EFT1

295
296                                     ;ERROR  141     RMCS1 NOT CLEARED BY DRIVE CLEAR
297
    003200  074010                              EMT141
    003202  076346                              EHT1
    003204  076472                              EDT1
    003206  076562                              EFT1

298
```

```
299                                     ;ERROR  142     RMDS NOT CLEARED BY DRIVE CLEAR
300
    003210  074020                              EMT142
    003212  076346                              EHT1
    003214  076472                              EDT1
    003216  076562                              EFT1

301
302                                     ;ERROR  143     RMER1 NOT CLEARED BY DRIVE CLEAR
303
    003220  074030                              EMT143
    003222  076346                              EHT1
    003224  076472                              EDT1
    003226  076562                              EFT1

304
305                                     ;ERROR  144     RMAS NOT CLEARED BY DRIVE CLEAR
306
    003230  074040                              EMT144
    003232  076346                              EHT1
    003234  076472                              EDT1
    003236  076562                              EFT1

307
308                                     ;ERROR  145     RMMR1  NOT CLEARED BY DRIVE CLEAR
309
    003240  074050                              EMT145
    003242  076346                              EHT1
    003244  076472                              EDT1
    003246  076562                              EFT1

310
311                                     ;ERROR  146     RMMR2 NOT CLEARED BY DRIVE CLEAR
312
    003250  074060                              EMT146
    003252  076346                              EHT1
    003254  076472                              EDT1
    003256  076562                              EFT1

313
314                                     ;ERROR  147     RMER2 NOT CLEARED BY DRIVE CLEAR
315
    003260  074070                              EMT147
    003262  076346                              EHT1
    003264  076472                              EDT1
    003266  076562                              EFT1

316
317                                     ;ERROR  150     RMEC2 NOT CLEARED BY DRIVE CLEAR
318
    003270  074100                              EMT150
    003272  076346                              EHT1
    003274  076472                              EDT1
    003276  076562                              EFT1

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

```
321
    003300  074110                      EMT151
    003302  076346                      EHT1
    003304  076472                      EDT1
    003306  076562                      EFT1

322
323                             ;ERROR  152     DRIVE FAULT
324
    003310  074122                      EMT152
    003312  076346                      EHT1
    003314  076472                      EDT1
    003316  076562                      EFT1

325
326                             ;ERROR  153     UNSAFE SHOULD BE SET BECAUSE DVC IS SET
327
    003320  074134                      EMT153
    003322  076346                      EHT1
    003324  076472                      EDT1
    003326  076562                      EFT1

328
329                             ;ERROR  154     UNSAFE SHOULD NOT BE SET, AC IS LOW
330
    003330  074152                      EMT154
    003332  076346                      EHT1
    003334  076472                      EDT1
    003336  076562                      EFT1

331
332                             ;ERROR  155     VOLUME VALID NOT SET BY PACK ACK
333
    003340  074170                      EMT155
    003342  076346                      EHT1
    003344  076472                      EDT1
    003346  076562                      EFT1

334
335                             ;ERROR  156     OFFSET MODE NOT SET BY OFFSET COMMAND
336
    003350  074202                      EMT156
    003352  076346                      EHT1
    003354  076472                      EDT1
    003356  076562                      EFT1

337
338                             ;ERROR  157     OFFSET MODE NOT RESET BY RTC COMMAND
339
    003360  074214                      EMT157
    003362  076346                      EHT1
    003364  076472                      EDT1
    003366  076562                      EFT1

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

```
003370  074226                  EMT160
003372  076346                  EHT1
003374  076472                  EDT1
003376  076562                  EFT1

343
344                     ;ERROR  161     RMDA NOT RESET BY RIP COMMAND
345
003400  074236                  EMT161
003402  076346                  EHT1
003404  076472                  EDT1
003406  076562                  EFT1

346
347                     ;ERROR  162     RMDC NOT RESET BY RIP COMMAND
348
003410  074250                  EMT162
003412  076346                  EHT1
003414  076472                  EDT1
003416  076562                  EFT1

349
350                     ;ERROR 163      DATA WAS ECC CORRECTED BUT DOES NOT COMPARE WITH
351                     ;               WRITE BUFFER
352
003420  076142                  EMT336
003422  076430                  EHT336
003424  076526                  EDT336
003426  076616                  EFT336

353
354                     ;ERROR  164     OPI SHOULD NOT BE SET
355
003430  074272                  EMT164
003432  076346                  EHT1
003434  076472                  EDT1
003436  076562                  EFT1

356
357                     ;ERROR  165     IVC SHOULD NOT BE SET
358
003440  074300                  EMT165
003442  076346                  EHT1
003444  076472                  EDT1
003446  076562                  EFT1

359
360                     ;ERROR  166     IAE SHOULD NOT BE SET
361
003450  074306                  EMT166
003452  076346                  EHT1
003454  076472                  EDT1
003456  076562                  EFT1

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

```
003460  074314                          EMT167
003462  076346                          EHT1
003464  076472                          EDT1
003466  076562                          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  074332                          EMT171
003502  076346                          EHT1
003504  076472                          EDT1
003506  076562                          EFT1

371
372                             ;ERROR  172     "ATA" NOT SET BY OFFSET COMMAND
373
003510  074342                          EMT172
003512  076346                          EHT1
003514  076472                          EDT1
003516  076562                          EFT1

374
375                             ;ERROR  173     RMER2 NOT INITIALIZED BY UNIBUS INIT
376
003520  074352                          EMT173
003522  076346                          EHT1
003524  076472                          EDT1
003526  076562                          EFT1

377
378                             ;ERROR  174     RMER2 NOT INITIALIZED BY CONTROLLER CLEAR
379
003530  074362                          EMT174
003532  076346                          EHT1
003534  076472                          EDT1
003536  076562                          EFT1

380
381                             ;ERROR  175     SELECTED DEVICE IS IN WRITE PROTECT
382
003540  074374                          EMT175
003542  076346                          EHT1
00354.  076472                          EDT1
003546  076562                          EFT1

383
384                             ;ERROR  176     CANNOT SET DIAGNOSTIC MODE
385
003550  074402                          EMT176
```

```
003552  076346                          EHT1
003554  076472                          EDT1
003556  076562                          EFT1

386
387                             ;ERROR  177     INCORRECT 'MOL' STATUS DURING DIAGNOSTIC MODE
388
003560  074410                          EMT177
003562  076346                          EHT1
003564  076472                          EDT1
003566  076562                          EFT1

389
390                             ;ERROR  200     INCORRECT 'PIP' STATUS DURING DIAGNOSTIC MODE
391
003570  074422                          EMT200
003572  076346                          EHT1
003574  076472                          EDT1
003576  076562                          EFT1

392
393                             ;ERROR  201     INCORRECT 'WRL' STATUS DURING DIAGNOSTIC MODE
394
003600  074434                          EMT201
003602  076346                          EHT1
003604  076472                          EDT1
003606  076562                          EFT1

395
396                             ;ERROR  202     INCORRECT 'SKI' STATUS DURING DIAGNOSTIC MODE
397
003610  074446                          EMT202
003612  076346                          EHT1
003614  076472                          EDT1
003616  076562                          EFT1

398
399                             ;ERROR  203     INCORRECT 'DVC' STATUS DURING DIAGNOSTIC MODE
400
003620  074460                          EMT203
003622  076346                          EHT1
003624  076472                          EDT1
003626  076562                          EFT1

401
402                             ;ERROR  204     'VV' WAS NOT RESET BY MAINTENANCE UNIT READY
403
003630  074472                          EMT204
003632  076346                          EHT1
003634  076472                          EDT1
003636  076562                          EFT1

404
405                             ;ERROR  205     SELECTED DEVICE HAS A PERSISTENT 'SKI' ERROR
406
003640  074510                          EMT205
003642  076346                          EHT1
```

```
                003644  076472                          EDT1
                003646  076562                          EFT1

    407
    408                                 ;ERROR  206     'LBC'' DID NOT SET DURING DIAGNOSTIC MODE
    409
                003650  074520                          EMT206
                003652  076346                          EHT1
                003654  076472                          EDT1
                003656  076562                          EFT1

    410
    411                                 ;ERROR  207     UNEXPECTED LOSS OF 'MOL'' - MEDIUM IS OFF LINE
    412
                003660  074530                          EMT207
                003662  076346                          EHT1
                003664  076472                          EDT1
                003666  076562                          EFT1

    413
    414                                 ;ERROR  210     UNEXPECTED LOSS OF VOLUME VALID - 'VV'' = 0
    415
                003670  074542                          EMT210
                003672  076346                          EHT1
                003674  076472                          EDT1
                003676  076562                          EFT1

    416
    417                                 ;ERROR  211     UNEXPECTED MECHANICAL MOTION - 'PIP'' = 1
    418
                003700  074550                          EMT211
                003702  076346                          EHT1
                003704  076472                          EDT1
                003706  076562                          EFT1

    419
    420                                 ;ERROR  212     UNEXPECTED DEVICE FAULT - 'DVC'' = 1
    421
                003710  074564                          EMT212
                003712  076346                          EHT1
                003714  076472                          EDT1
                003716  076562                          EFT1

    422
    423                                 ;ERROR  213     UNEXPECTED SEEK INCOMPLETE ERROR - 'SKI'' = 1
    424
                003720  074600                          EMT213
                003722  076346                          EHT1
                003724  076472                          EDT1
                003726  076562                          EFT1

    425
    426                                 ;ERROR  214     DRIVE EXECUTED A RECALIBRATE WITH ERROR SET
    427
                003730  074610                          EMT214
                003732  076360                          EHT2
                003734  076502                          EDT2
```

```
        003736  076572                          EFT2

    428
    429                                 ;ERROR  215     DRIVE DID NOT DETECT ''IVC'' ERROR DURING RECALIBRATE
    430
        003740  074630                          EMT215
        003742  076360                          EHT2
        003744  076502                          EDT2
        003746  076572                          EFT2

    431
    432                                 ;ERROR  216     INCORRECT ''IVC'' STATUS
    433
        003750  074642                          EMT216
        003752  076346                          EHT1
        003754  076472                          EDT1
        003756  076562                          EFT1

    434
    435                                 ;ERROR  217     INCORRECT ''IAE'' STATUS
    436
        003760  074652                          EMT217
        003762  076346                          EHT1
        003764  076472                          EDT1
        003766  076562                          EFT1

    437
    438                                 ;ERROR 220      INCORRECT 'WLE'' STATUS
    439
        003770  074662                          EMT220
        003772  076346                          EHT1
        003774  076472                          EDT1
        003776  076562                          EFT1

    440
    441                                 ;ERROR  221     INCORRECT 'DPI'' STATUS
    442
        004000  074672                          EMT221
        004002  076346                          EHT1
        004004  076472                          EDT1
        004006  076562                          EFT1

    443
    444                                 ;ERROR  222     RM DID NOT DETECT RMR ERROR
    445
        004010  074702                          EMT222
        004012  076346                          EHT1
        004014  076472                          EDT1
        004016  076562                          EFT1

    446
    447                                 ;ERROR  223     RM DID NOT DETECT PARITY ERROR ON MASSBUS CONTROL BUS
    448
        004020  074712                          EMT223
        004022  076404                          EHT223
        004024  076516                          EDT223
        004026  076606                          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
```

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

                                    ;ERROR  243     UNUSED
004220  000000                              0
004222  000000                              0
004224  000000                              0
004226  000000                              0

                                    ;ERROR  244     UNUSED
004230  000000                              0
004232  000000                              0
004234  000000                              0
004236  000000                              0

                                    ;ERROR  245     UNUSED
004240  000000                              0
004242  000000                              0
004244  000000                              0
004246  000000                              0

                                    ;ERROR  246     ''ATA'' NOT RESET BY GO WHEN 'ERR'' = 0
004250  074766                              EMT246
004252  076346                              EHT1
004254  076472                              EDT1
004256  076562                              EFT1

                                    ;ERROR  247     ''ATA'' NOT RESET BY WRITING RMAS
004260  074776                              EMT247
004262  076346                              EHT1
004264  076472                              EDT1
004266  076562                              EFT1

                                    ;ERROR  250     ''ATA'' WAS RESET BY GO WHEN 'ERR'' = 1
004270  075010                              EMT250
004272  076346                              EHT1
004274  076472                              EDT1
004276  076562                              EFT1

                                    ;ERROR  251     PROGRAM INTERRUPT WAS NOT GENERATED
```

```
514
     004300  075024                          EMT251
     004302  076360                          EHT2
     004304  076502                          EDT2
     004306  076572                          EFT2

515
516                                 ;ERROR  252     PROGRAM INTERRUPT SHOULD NOT HAVE BEEN GFNERATED
517
     004310  075032                          EMT252
     004312  076360                          EHT2
     004314  076502                          EDT2
     004316  076572                          EFT2

518
519                                 ;ERROR  253     OFFSET MODE WAS NOT RESET BY WRITING RMDC
520
     004320  075040                          EMT253
     004322  076346                          EHT1
     004324  076472                          EDT1
     004326  076562                          EFT1

521
522                                 ;ERROR  254     INCORRECT ''ILF'' STATUS
523
     004330  075056                          EMT254
     004332  076346                          EHT1
     004334  076472                          EDT1
     004336  076562                          EFT1

524
525                                 ;ERROR  255     INCORRECT ''ATA'' STATUS
526
     004340  075066                          EMT255
     004342  076346                          EHT1
     004344  076472                          EDT1
     004346  076562                          EFT1

527
528                                 ;ERROR  256     INCORRECT ''ILR'' STATUS
529
     004350  075076                          EMT256
     004352  076416                          EHT256
     004354  076516                          EDT223
     004356  076606                          EFT223

530
531                                 ;ERROR  257     INVALID IAE STATUS DURING SEARCH COMMAND
532
     004360  075106                          EMT257
     004362  076346                          EHT1
     004364  076472                          EDT1
     004366  076562                          EFT1

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

```
004370  075120                          EMT260
004372  076346                          EHT1
004374  076472                          EDT1
004376  076562                          EFT1

536
537                             ;ERROR  261     DRIVE EXECUTED SEARCH WITH ERROR SET
538
004400  075132                          EMT261
004402  076346                          EHT1
004404  076472                          EDT1
004406  076562                          EFT1

539
540                             ;ERROR  262     'LBC' ERROR NOT SET DURING DIAGNOSTIC MODE
541
004410  075152                          EMT262
004412  076346                          EHT1
004414  076472                          EDT1
004416  076562                          EFT1

542
543                             ;ERROR  263     'SKI' ERROR DURING SEARCH COMMAND
544
004420  075162                          EMT263
004422  076346                          EHT1
004424  076472                          EDT1
004426  076562                          EFT1

545
546                             ;ERROR  264     'IVC' ERROR DURING SEARCH - LOST VOLUME VALID
547
004430  075172                          EMT264
004432  076346                          EHT1
004434  076472                          EDT1
004436  076562                          EFT1

548
549                             ;ERROR  265     ERRONEOUS IVC ERROR DURING SEARCH - VOLUME IS VALID
550
004440  075212                          EMT265
004442  076346                          EHT1
004444  076472                          EDT1
004446  076562                          EFT1

551
552                             ;ERROR  266     DEVICE FAULT (DVC) DURING SEARCH
553
004450  075232                          EMT266
004452  076346                          EHT1
004454  076472                          EDT1
004456  076562                          EFT1

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

ERROR POINTER TABLE

```
004460  075244                          EMT267
004462  076346                          EHT1
004464  076472                          EDT1
004466  076562                          EFT1

558
559                             ;ERROR  270     OPI ERROR DURING SEARCH BECAUSE MOL = 0
560
004470  075262                          EMT270
004472  076346                          EHT1
004474  076472                          EDT1
004476  076562                          EFT1

561
562                             ;ERROR  271     OPI ERROR DURING SEARCH BECAUSE ON CYLINDER
563                             ;               DIDN'T DROP
564
004500  075276                          EMT271
004502  076346                          EHT1
004504  076472                          EDT1
004506  076562                          EFT1

565
566                             ;ERROR  272     LOST MOL DURING SEARCH, OPI IS NOT SET
567
004510  075314                          EMT272
004512  076346                          EHT1
004514  076472                          EDT1
004516  076562                          EFT1

568
569                             ;ERROR  273     PIP STIL SET AFTER SEARCH
570
004520  075332                          EMT273
004522  076346                          EHT1
004524  076472                          EDT1
004526  076562                          EFT1

571
572                             ;ERROR  274     PARITY ERROR OCCURRED WHILE WRITING REMOTE
573                             ;               REGISTERS BUT MXF DID NOT SET
574
004530  075350                          EMT274
004532  076346                          EHT1
004534  076472                          EDT1
004536  076562                          EFT1

575
576                             ;ERROR  275     MXF ERROR - COMPOSITE ERROR OCCURRED BEFORE DATA
577                             ;               COMMAND STARTED
578
004540  075366                          EMT275
004542  076346                          EHT1
004544  076472                          EDT1
004546  076562                          EFT1

579
```

```
                                        ;ERROR  276     'OPI'' ERROR DURING DATA TRANSFER BECAUSE 'MOL'' WAS
                                        ;               ZERO
004550  075400                                  EMT276
004552  076346                                  EHT1
004554  076472                                  EDT1
004556  076562                                  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  075414                                  EMT277
004562  076346                                  EHT1
004564  076472                                  EDT1
004566  076562                                  EFT1

                                        ;ERROR  300     ''IVC'' ERROR DURING DATA TRANSFER BECAUSE VOLUME
                                        ;               WAS NOT VALID
004570  075432                                  EMT300
004572  076346                                  EHT1
004574  076472                                  EDT1
004576  076562                                  EFT1

                                        ;ERROR  301     ERRONEOUS ''IVC'' ERROR DURING DATA TRANSFER - VOLUME
                                        ;               IS VALID
004600  075452                                  EMT301
004602  076346                                  EHT1
004604  076472                                  EDT1
004606  076562                                  EFT1

                                        ;ERROR  302     ''ILR'' ERROR DURING DATA TRANSFER
004610  075474                                  EMT302
004612  076346                                  EHT1
004614  076472                                  EDT1
004616  076562                                  EFT1

                                        ;ERROR  303     ''ILF'' ERROR DURING DATA TRANSFER
004620  075506                                  EMT303
004622  076346                                  EHT1
004624  076472                                  EDT1
004626  076562                                  EFT1

                                        ;ERROR  304     'RMR'' ERROR DURING DATA TRANSFER
004630  075520                                  EMT304
004632  076346                                  EHT1
```

```
004634  076472                          EDT1
004636  076562                          EFT1

605
606                             ;ERROR  305     INCORRECT ''IAE'' STATUS DURING DATA TRANSFER
607
004640  075532                          EMT305
004642  076346                          EHT1
004644  076472                          EDT1
004646  076562                          EFT1

608
609                             ;ERROR  306     ''SKI'' ERROR DURING DATA TRANSFER
610
004650  075544                          EMT306
004652  076346                          EHT1
004654  076472                          EDT1
004656  076562                          EFT1

611
612                             ;ERROR  307     DRIVE DID NOT DETECT SKI ERROR DUE TO CYLINDER
613
004660  075554                          EMT307
004662  076346                          EHT1
004664  076472                          EDT1
004666  076562                          EFT1

614
615                             ;ERROR  310     DEVICE FAULT DURING DATA TRANSFER
616
004670  075574                          EMT310
004672  076346                          EHT1
004674  076472                          EDT1
004676  076562                          EFT1

617
618                             ;ERROR  311     LOSS OF BIT CLOCK DURING DATA TRANSFER
619
004700  075606                          EMT311
004702  076346                          EHT1
004704  076472                          EDT1
004706  076562                          EFT1

620
621                             ;ERROR  312     LOSS OF SYSTEM CLOCK DURING DATA TRANSFER
622
004710  075620                          EMT312
004712  076346                          EHT1
004714  076472                          EDT1
004716  076562                          EFT1

623
624                             ;ERROR  313     UNSAFE ERROR DURING DATA TRANSFER (DVC = 0)
625
004720  075632                          EMT313
004722  076346                          EHT1
004724  076472                          EDT1
```

```
004726  076562                          EFT1

626
627                             ;ERROR  314     DRIVE TIMING ERROR DURING DATA TRANSFER
628
004730  075652                          EMT314
004732  076346                          EHT1
004734  076472                          EDT1
004736  076562                          EFT1

629
630                             ;ERROR  315     WRITE LOCK ERROR
631
004740  075664                          EMT315
004742  076346                          EHT1
004744  076472                          EDT1
004746  076562                          EFT1

632
633                             ;ERROR  316     ERRONEOUS WRITE LOCK ERROR
634
004750  075676                          EMT316
004752  076346                          EHT1
004754  076472                          EDT1
004756  076562                          EFT1

635
636                             ;ERROR  317     HEADER CRC ERROR DURING DATA TRANSFER
637
004760  075710                          EMT317
004762  076346                          EHT1
004764  076472                          EDT1
004766  076562                          EFT1

638
639                             ;ERROR  320     FORMAT ERROR DURING DATA TRANSFER
640
004770  075720                          EMT320
004772  076346                          EHT1
004774  076472                          EDT1
004776  076562                          EFT1

641
642                             ;ERROR  321     HEADER COMPARE ERROR DURING DATA TRANSFER
643
005000  075730                          EMT321
005002  076346                          EHT1
005004  076472                          EDT1
005006  076562                          EFT1

644
645                             ;ERROR  322     HEADER ERRORS SHOULD NOT BE SET
646
005010  075740                          EMT322
005012  076346                          EHT1
005014  076472                          EDT1
005016  076562                          EFT1
```

```
647
648                                     ;ERROR  323     DATA CHECK ERROR DURING DATA TRANSFER
649
     005020  075746                               EMT323
     005022  076346                               EHT1
     005024  076472                               EDT1
     005026  076562                               EFT1

650
651                                     ;ERROR  324     CORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
652
     005030  075756                               EMT324
     005032  076346                               EHT1
     005034  076472                               EDT1
     005036  076562                               EFT1

653
654                                     ;ERROR  325     UNCORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
655
     005040  075770                               EMT325
     005042  076346                               EHT1
     005044  076472                               EDT1
     005046  076562                               EFT1

656
657                                     ;ERROR  326     DATA PARITY ERROR DURING READ COMMMAND
658
     005050  076002                               EMT326
     005052  076346                               EHT1
     005054  076472                               EDT1
     005056  076562                               EFT1

659
660                                     ;ERROR  327     OFFSET MODE NOT RESET BY WRITE COMMAND
661
     005060  076020                               EMT327
     005062  076346                               EHT1
     005064  076472                               EDT1
     005066  076562                               EFT1

662
663                                     ;ERROR  330     DATA PARITY ERROR DURING WRITE COMMAND
664
     005070  076032                               EMT330
     005072  076346                               EHT1
     005074  076472                               EDT1
     005076  076562                               EFT1

665
666                                     ;ERROR  331     WRITE CLOCK FALURE DURING WRITE COMMAND
667
     005100  076042                               EMT331
     005102  076346                               EHT1
     005104  076472                               EDT1
     005106  076562                               EFT1
```

```
668
669                             ;ERROR  332     DATA LATE ERROR DURING DATA TRANSFER
670
     005110  076054                             EMT332
     005112  076346                             EHT1
     005114  076472                             EDT1
     005116  076562                             EFT1

671
672                             ;ERROR  333     PIP STIL SET AFTER DATA TRANSFER - SKI = 0
673
     005120  076066                             EMT333
     005122  076346                             EHT1
     005124  076472                             EDT1
     005126  076562                             EFT1

674
675                             ;ERROR  334     LOST MOL DURING DATA TRANSFER - OPI = 0
676
     005130  076104                             EMT334
     005132  076346                             EHT1
     005134  076472                             EDT1
     005136  076562                             EFT1

677
678                             ;ERROR  335     LOST VOLUME VALID DURING DATA TRANSFER - IVC  = 0
679
     005140  076122                             EMT335
     005142  076346                             EHT1
     005144  076472                             EDT1
     005146  076562                             EFT1

680
681                             ;ERROR  336     DATA READ DOES NOT COMPARE WITH DATA WRITTEN
682
     005150  076142                             EMT336
     005152  076430                             EHT336
     005154  076526                             EDT336
     005156  076616                             EFT336

683
684                             ;ERROR  337     WRITE CHECK ERROR NOT DETECTED
685
     005160  076152                             EMT337
     005162  076442                             EHT337
     005164  076536                             EDT337
     005166  076626                             EFT337

686
687                             ;ERROR  340     WRITE CHECK ERROR AT UNEXPECTED ADDRESS
688
     005170  076162                             EMT340
     005172  076430                             EHT336
     005174  076526                             EDT336
     005176  076616                             EFT336

689
```

```
690                                     ;ERROR  341     INCORRECT DATA DURING WRITE CHECK ERROR
691
     005200  076174                               EMT341
     005202  076430                               EHT336
     005204  076526                               EDT336
     005206  076616                               EFT336

692
693                                     ;ERROR  342     ''IVC'' ERROR NOT DETECTED DURING DATA TRANSFER
694
     005210  076202                               EMT342
     005212  076346                               EHT1
     005214  076472                               EDT1
     005216  076562                               EFT1

695
696                                     ;ERROR  343     ''FER'' NOT DETECTED DURING DATA TRANSFER
697
     005220  076214                               EMT343
     005222  076346                               EHT1
     005224  076472                               EDT1
     005226  076562                               EFT1

698
699                                     ;ERROR  344     ''HCE'' NOT DETECTED DURING DATA TRANSFER
700
     005230  076226                               EMT344
     005232  076454                               EHT344
     005234  076546                               EDT344
     005236  076636                               EFT344

701
702                                     ;ERROR  345     ''BSE'' NOT DETECTED DURING DATA TRANSFER
703
     005240  076240                               EMT345
     005242  076346                               EHT1
     005244  076472                               EDT1
     005246  076562                               EFT1

704
705                                     ;ERROR  346     HEADER ERROR WAS DETECTED W/ HCI SET
706
     005250  076250                               EMT346
     005252  076346                               EHT1
     005254  076472                               EDT1
     005256  076562                               EFT1

707
708                                     ;ERROR  347     DATA TRANSFER NOT ABORTED W/ COMP ERROR SET
709
     005260  076264                               EMT347
     005262  076346                               EHT1
     005264  076472                               EDT1
     005266  076562                               EFT1

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

```
712
    005270  076276                      EMT350
    005272  076346                      EHT1
    005274  076472                      EDT1
    005276  076562                      EFT1

713
714                             ;ERROR  351     "ATA" DID NOT SET DURING SEARCH
715
    005300  076314                      EMT351
    005302  076346                      EHT1
    005304  076472                      EDT1
    005306  076562                      EFT1

716
717                             ;ERROR  352     PROGRAM TIMEOUT WHILE TESTING RMLA
718
    005310  076324                      EMT352
    005312  000000                      0
    005314  000000                      0
    005316  000000                      0

719
720                             ;ERROR  353     LOOK AHEAD TEST FAILS
721
    005320  076330                      EMT353
    005322  076466                      EHT353
    005324  076560                      EDT353
    005326  076646                      EFT353

722
723                             ;ERROR  354     BSE SHOULD NOT BE SET
724
    005330  076340                      EMT354
    005332  076346                      EHT1
    005334  076472                      EDT1
    005336  076562                      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$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
005412                          ;;$:
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  065342  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005460  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005466  012737  065736  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005474  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005502  012737  067476  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005510  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005516  012737  067604  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005524  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005532  013737  037514  037506          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     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
005702                          67$:
 20                             ;SETUP ''TIMEOUT'' TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
 21 005702  012737  005340  000004        MOV     #BADTMO,ERRVEC  ;SETUP FOR UNEXPECTED TIMEOUT
 22 005710  012737  000300  000006        MOV     #PR6,ERRVEC+2   ;LEVEL 6
 23 005716  012746  000300                MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
005722  012746  005730                    MOV     #68$,-(SP)      ;;PUT NEW PC ON  STACK
005726  000002                          RTI                     ;;POP NEW PC AND PS
005730                          68$:
 24
 25                             .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  037650  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$:  .ASCIZ  <CRLF>@CZRMOA0 - RM05/3/2 FUNCTIONAL TEST, PART 3@<CRLF>
006072                          69$:
 26
 27
 28                             ;THE FOLLOWING FINDS OUT THE PROGRAM CONTROL MODE:
 29                             ;PAPER TAPE (MANUAL), ACT11, XXDP CHAIN OR DUMP
 30 006072  005037  001330              CLR     XXDP            ;CLEAR 'XXDP' LOAD DEVICE STORAGE
 31 006076  122737  000016  000041      CMPB    #16,@#41        ;LOADED FROM AN RM05/3/2 ?
 32 006104  001160                      BNE     3$              ;BRANCH IF NOT
 33 006106  013737  000040  001330      MOV     @#40,XXDP       ;GET DEVICE INDICATOR AND NUMBER
 34 006114  122737  000007  001330      CMPB    #7,XXDP         ;IS IT A VALID NUMBER ?
 35 006122  103002                      BHIS    1$              ;YES
 36 006124  105037  001330              CLRB    XXDP            ;NO, DEFAULT TO DRIVE 0
 37 006130  005737  000042        1$:   TST     @#42            ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
 38 006134  001425                      BEQ     2$              ;BR IF NEITHER
 39 006136  104401  006144              TYPE    ,74$            ;;TYPE ASCIZ STRING
006142  000412                          BR      73$             ;;GET OVER THE ASCIZ
                                ;;74$:  .ASCIZ  <CRLF>/NOT TESTING DRIVE /
006170                          73$:
 40 006170  005046                      CLR     -(SP)           ;CLEAR WORD ON STACK
 41 006172  113716  001330              MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
 42 006176  104403                      TYPOS                   ;TYPE THE ADDRESS
 43 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
     62 006446  105737  001150        ;CHECK FOR AUTO MODE OR STANDLONE MODE
                                                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     $BASE,R0        ;LOAD THE BASE ADDRESS
     73
     74 006500  136137  071616  001300  1$:     BITB    ATNTBL(R1),$DEVM        ;IS DEVICE PRESENT IN MAP ?
     75 006506  001462                          BEQ     6$              ;BR IF NO
     76 006510  012737  071416  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$:     MOV     #CLR,RMCS2(R0)  ;CLEAR MASS BUS
     83 006540  010160  000010                  MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
     84 006544  005760  000012                  TST     RMDS(R0)        ;TRY TO ACCESS AN RM DRIVE REGISTER
     85 006550  012737  071436  006652          MOV     #NOTPRS,5$      ;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  071453  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  071472  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  071616  001300  3$:     BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
     97 006632  104401  001217          4$:     TYPE    ,$CRLF          ;CR-LF
     98 006636  104401  071410                  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  066146                          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$:
006720  104401  070755                          TYPE    ,CNSL00         ;MAINTAIN PREVIOUS PARAMETERS ?
006724  104411                                  RDCHR                   ;GET RESPONSE
006726  012637  001176                          MOV     (SP)+,$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  071352                          TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT '
006772  000752                                  BR      1$              ;TRY AGAIN
                                        
                                        ;SEE IF OPERATOR WANTS HELP TEXT
006774                                  3$:
006774  104401  070345                          TYPE    ,MSHELP         ;WANT HELP ?
007000  104411                                  RDCHR                   ;GET RESPONSE
007002  012637  001176                          MOV     (SP)+,$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  071352                          TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT'
007036  000756                                  BR      3$              ;TRY AGAIN
007040  104401  106314                  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  070722                          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  071352                          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  071010                          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)
```

```
                                                TYPE    ,QUES           ;TYPE '' ? ''
                                                RDOCT                   ;GET NEW BUS ADDRESS
                                                MOV     (SP)+,$TMP1     ;CARRIAGE RETURN ?
                                                BEQ     8$              ;YES-SKIP TO NEXT ENTRY
                                                CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE ?
                                                BLOS    7$              ;YES
                                                TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
                                                BR      6$+4            ;TRY AGAIN
                                        7$:     MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS

                                        8$:     TYPE    ,CNSL03
                                                CLR     -(SP)
                                                MOVB    $VECT1,(SP)     ;GET CURRENT VECTOR ADDRESS
                                                TYPOS
                                                .BYTE   3               ;TYPE 3 DIGITS
                                                .BYTE   0               ;SUPPRESS LEADING ZEROS
                                                TYPE    ,QUES           ;TYPE '' ? ''
                                                RDOCT                   ;GET NEW VECTOR ADDRESS
                                                MOV     (SP)+,$TMP1     ;CARRIAGE RETURN?
                                                BEQ     10$             ;YES-SKIP TO NEXT ENTRY
                                                CMP     #1000,$TMP1     ;VECTOR ADDRESS < 1000 ?
                                                BHI     9$              ;YES!!
                                                TYPE    ,CNSL04         ;TYPE WARNING MESSAGE
                                                BR      8$              ;RETRY
                                        9$:     MOVB    $TMP1,$VECT1    ;STORE NEW VECTOR ADDRESS

                                        10$:    TYPE    ,CNSL05
                                                CLR     -(SP)
                                                MOVB    $VECT1+1,(SP)   ;GET CURRENT BR LEVEL
                                                ASR     (SP)
                                                ASR     (SP)
                                                ASR     (SP)
                                                ASR     (SP)
                                                TYPOS
                                                .BYTE   1               ;ONLY 1 DIGIT
                                                .BYTE   0               ;SUPPRESS LEADING ZEROS
                                                TYPE    ,QUES
                                                RDOCT                   ;GET NEW PRIORITY
                                                MOV     (SP)+,$TMP1     ;CARRIAGE RETURN ?
                                                BEQ     12$             ;YES-SKIP TO NEXT ENTRY
                                                CMP     $TMP1,#7        ;LEGAL PRIORITY ?
                                                BLT     11$             ;YES!!
                                                TYPE    ,CNSL06         ;TYPE WARNING MESSAGE
                                                BR      10$             ;TRY AGAIN
                                        11$:    ASL     $TMP1           ;STORE NEW PRIORITY
                                                ASL     $TMP1
                                                ASL     $TMP1
                                                ASL     $TMP1
                                                ASL     $TMP1
                                                MOVB    $TMP1,$VECT1+1

                                        ;DIALOGUE TO INPUT DEVICE NUMBERS
                                        12$:
                                                INC     #-1             ;FIRST TIME THRU ?
                                                BNE     13$             ;BR IF NO
                                                TYPE    ,CNSL07         ;TYPE INPUT INSTRUCTIONS
```

```
007372  104401  001217          13$:    TYPE    ,$CRLF          ;CR-LF
007376  005037  001300          14$:    CLR     $DEVM           ;CLEAR DEVICE MAP
007402  104401  071374                  TYPE    ,MSDRVS         ;TYPE 'DRIVE(S): '
007406  104411                          RDCHR
007410  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
007414  023727  001176  000101          CMP     $TMP1,#'A       ;IS INPUT ''A'' ?
007422  001007                          BNE     15$             ;NO
007424  104401  070331                  TYPE    ,ALL            ;YES, TYPE ''ALL'' AND GO
007430  012737  000377  001300          MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
007436  000137  006462                  JMP     XSIZ            ;AUTO SIZE.

007442  023727  001176  000015  15$:    CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
007450  001436                          BEQ     17$             ;YES
007452  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
007456  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
007464  002430                          BLT     17$             ;YES
007466  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
007474  003427                          BLE     18$             ;NO
007476  000423                          BR      17$             ;ILLEGAL INPUT

007500  104411                  16$:    RDCHR
007502  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
007506  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
007514  001432                          BEQ     19$             ;YES
007516  104401  070342                  TYPE    ,COMMA          ;TYPE ', '
007522  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
007526  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
007534  002404                          BLT     17$             ;YES
007536  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
007544  003403                          BLE     18$             ;NO
007546  104401  071352          17$:    TYPE    ,CNSL08         ;TYPE '' ?ILLEGAL INPUT''
007552  000711                          BR      14$             ;RETRY

007554  013701  001176          18$:    MOV     $TMP1,R1        ;R1 = DRIVE NUMBER
007560  042701  177770                  BIC     #^C7,R1
007564  156137  071616  001300          BISB    ATNTBL(R1),$DEVM        ;SET DEVICE IN MAP
007572  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ?
007600  101337                          BHI     16$             ;NO
007602  104401  001217          19$:    TYPE    ,$CRLF          ;CR-LF
007606  000137  006462                  JMP     XSIZ            ;GO SIZE DEVICES
```

```
                                ;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  071616  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  044720                  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  037460          1$:     JMP     $EOP            ;RETURN CONTROL TO MONITOR

010012  105037  001116          2$:     CLRB    $TSTNM          ;RESET TEST NUMBER
010016  005037  001206                  CLR     $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
010022  005037  001326                  CLR     CTLFG           ;CLEAR CONTROL-C FLAG
010026  004737  066146                  JSR     PC,$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$:
    42  010044  117737  171414  001234          MOVB    @TSTQUE,$UNIT   ;LOAD UNIT NUMBER
    43  010052  005037  001510                  CLR     MEDENB          ;CLEAR MEDIA ENABLE
    44
    45                                  ;CLEAR MASSBUS CONTROLLER, SELECT DRIVE AND DETERMINE THE LAST TRACK
    46                                  ;OF THE DIFFERENT DRIVE TYPES
    47  010056  012737  002000  001332          MOV     #TA4,LSTRK      ;ASSUME LAST TRACK FOR RM02/3 = 4.
    48  010064  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    49  010070  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
    50  010076  117760  171362  000010          MOVB    @TSTQUE,RMCS2(R0)       ;SELECT DEVICE UNDER TEST
    51  010104  016002  000026                  MOV     RMDT(R0),R2     ;GET RMDT AND
    52  010110  042702  177770                  BIC     #177770,R2      ;SAVE DRIVE TYPE BITS
    53  010114  022702  000007                  CMP     #7,R2           ;IS IT AN RM05 ?
    54  010120  001003                          BNE     3$              ;NO, MUST BE AN RM02 OR RM03
    55  010122  012737  011000  001332          MOV     #TA16!TA2,LSTRK ;YES--SET LAST TRACK = 18.
    56  010130                          3$:
    57
```

```
                                ;;*****************************************************************
                                ;*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  037460 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  053604         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
     37 010360                          2$:
     38 010360  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
     39 010364  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
     40 010370  012737  010460  000004                  MOV     #5$,ERRVEC
     41 010376  013737  000300  000006                  MOV     PR6,ERRVEC+2
     42
     43 010404  016037  000000  001176                  MOV     RMCS1(R0),$TMP1 ;GET DVA STATUS
     44 010412  016037  000010  001174                  MOV     RMCS2(R0),$TMP0 ;GET NED STATUS
     45 010420  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
     46 010424  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     47 010430  032737  010000  001174                  BIT     #NED,$TMP0      ;NONEXISTENT DEVICE ?
     48 010436  001402                                  BEQ     3$              ;NO!!
     49 010440  104111                                  EMT     111
     50 010442  000414                                  BR      7$
     51 010444  032737  004000  001176  3$:             BIT     #DVA,$TMP1      ;DEVICE AVAILABLE ?
     52 010452  001012                                  BNE     9$              ;YES!!
     53 010454  104112                                  EMT     112
     54 010456  000406                                  BR      7$
     55
     56 010460  022626                  5$:             CMP     (SP)+,(SP)+     ;ADJUST STACK
     57 010462  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
     58 010466  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     59 010472  104113                                  EMT     113
     60 010474  000137  037422          7$:             JMP     $EOSP
     61 010500                          9$:
     62
     63                                 ;;*******************************************************************
                                        ;*TEST 3        DRIVE TYPE 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     $BASE,R0        ;R0 = UNIBUS ADDRESS
                010514  013701  001464          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                010520  012737  000003  001226  MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
     64
     65 010526  004737  053604                  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
     66 010544                          2$:
     67 010544  112737  000026  001522          MOVB    #RMDT,GETINX    ;SETUP GET INDEX TABLE
                010552  112737  000200  001523  MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
                010560  012737  010702  001362  MOV     #5$,RMDTI       ;SET RMDT INPUT BUFFER = 5$
                010566  004737  044230          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
     68
     69 010600  022737  020024  001362  3$:     CMP     #SNGPRT,RMDTI   ;SINGLE PORT RM03 ?
     70 010606  001435                          BEQ     5$              ;YES !!
     71 010610  022737  024024  001362          CMP     #DULPRT,RMDTI   ;DUAL PORT RM03 ?
     72 010616  001431                          BEQ     5$              ;YES !!
     73
```

```
010620  022737  020025  001362           CMP     #SNGPRT!BIT0,RMDTI       ;SINGLE PORT RM02 ?
010626  001425                           BEQ     5$                       ;YES !!
010630  022737  024025  001362           CMP     #DULPRT!BIT0,RMDTI       ;DUAL PORT RM02 ?
010636  001421                           BEQ     5$                       ;YES !!

010640  022737  020027  001362           CMP     #SNGPRT!BIT1!BIT0,RMDTI  ;SINGLE PORT RM05 ?
010646  001415                           BEQ     5$                       ;YES !!
010650  022737  024027  001362           CMP     #DULPRT!BIT1!BIT0,RMDTI  ;DUAL PORT RM05 ?
010656  001411                           BEQ     5$

010660  012737  020024  001176           MOV     #SNGPRT,$TMP1     ;LOAD ACCEPTABLE VALUES
010666  012737  024024  001200           MOV     #DULPRT,$TMP2
010674  104114                           EMT     114

010676  000137  037422          4$:      JMP     $EOSP             ;GO TO SUBPASS HANDLER.
010702                          5$:


                                ;;*************************************************************
                                ;*TEST 4         WRITE, READ 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

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

010730                          10$:


                                ;PREPARE DEVICE FOR TEST
010730  004737  040472                   JSR     PC,TSTPRP         ;PREPARE DEVICE FOR TEST
010734  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
010736  000404                           BR      20$               ;GO TO 20$ IF NO ERROR
010740  000240                           NOP                       ;RETURN HERE IF ERROR
010742  104000                           EMT                       ;ERROR # DEFINED BY TSTPRP SUBROUTINE
010744  000137  011710                   JMP     350$              ;GO TO 350$ IF ERROR


                                ;LOAD PARAMETERS AND GENERATE DATA BUFFER
010750                          20$:
010750  012737  000000  001444           MOV     #0,RMDC0                  ;CYLINDER = 0
010756  012737  000000  001416           MOV     #0,RMDA0                  ;TRACK = 0, SECTOR = 0
010764  012737  106314  001414           MOV     #BUFONE,RMBA0             ;BUS ADDRESS
010772  012737  177376  001412           MOV     #-258.,RMWC0              ;2 + 256. WORDS (2'S COMP)
011000  012737  010000  001442           MOV     #FMT16,RMOF0              ;16 BIT FORMAT
011006  012737  000063  001410           MOV     #WH!GO,RMCS10             ;WRITE HEADER AND DATA COMMAND
```

```
                                        ;VERIFY THAT SECTOR IS NOT BAD
011014  004737  041416                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
011020  000405                                BR      25$             ;GO TO 25$ IF NO ERROR
011022  104401  070230                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
011026  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
011030  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011034                                  25$:
011034  012737  071730  001174                MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
011042  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
011050  004737  043344                        JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
011054  012702  001551                        MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
011060  112722  000034                        MOVB    #RMDC,(R2)+
011064  112722  000006                        MOVB    #RMDA,(R2)+
011070  112722  000004                        MOVB    #RMBA,(R2)+
011074  112722  000002                        MOVB    #RMWC,(R2)+
011100  112722  000032                        MOVB    #RMOF,(R2)+
011104  112722  000000                        MOVB    #RMCS1,(R2)+
011110  112712  000200                        MOVB    #200,(R2)               ;TERMINATE TABLE
011114                                  30$:
011114  004737  044500                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011120  000404                                BR      40$             ;GO TO 40$ IF NO ERROR
011122  000240                                NOP                     ;RETURN HERE IF ERROR
011124  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
011126  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011132                                  40$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
011132  004737  044144                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
011136  004737  045042                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
011142  004737  044230                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
011146  000404                                BR      50$             ;GO TO 50$ IF NO ERROR
011150  000240                                NOP                     ;RETURN HERE IF ERROR
011152  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
011154  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011160                                  50$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
011160  004737  057742                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
011164  000405                                BR      60$             ;GO TO 60$ IF NO ERROR
011166  000240                                NOP                     ;RETURN HERE IF ERROR
011170  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
011172  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011174  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011200                                  60$:

                                        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
011200  012737  011210  001124                MOV     #70$,$LPERR
011206  000410                                BR      80$                     ;SKIP TO WRITE OPERATION

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

```
                                        70$:
                                        ;PREPARE DEVICE FOR TEST
011210  004737  040472                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
011214  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
011216  000404                                BR      80$             ;GO TO 80$ IF NO ERROR
011220  000240                                NOP                     ;RETURN HERE IF ERROR
011222  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
011224  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011230                                  80$:
                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
011230                                  120$:
011230  012737  106320  001414                MOV     #BUFONE+4,RMBAO         ;MOVE MEMORY ADDRESS
011236  012737  177400  001412                MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
011244  012737  000061  001410                MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
011252  012702  001551                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
011256  112722  000006                        MOVB    #RMDA,(R2)+
011262  112722  000034                        MOVB    #RMDC,(R2)+
011266  112722  000032                        MOVB    #RMOF,(R2)+
011272  112722  000004                        MOVB    #RMBA,(R2)+
011276  112722  000002                        MOVB    #RMWC,(R2)+
011302  112722  000000                        MOVB    #RMCS1,(R2)+
011306  112722  000200                        MOVB    #200,(R2)+              ;TERMINATE TABLE

011312  004737  044500                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011316  000404                                BR      130$            ;GO TO 130$ IF NO ERROR
011320  000240                                NOP                     ;RETURN HERE IF ERROR
011322  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
011324  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011330                                  130$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
011330  004737  044144                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
011334  004737  045042                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
011340  004737  044230                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
011344  000404                                BR      140$            ;GO TO 140$ IF NO ERROR
011346  000240                                NOP                     ;RETURN HERE IF ERROR
011350  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
011352  000137  011710                        JMP     350$            ;GO TO 350$ IF ERROR
011356                                  140$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
011356  004737  045226                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
011362  000405                                BR      150$            ;GO TO 150$ IF NO ERROR
```

```
        011364  000240                          NOP                     ;RETURN HERE IF ERROR
        011366  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        011370  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011372  000137  011710                  JMP     350$            ;GO TO 350$ IF ERROR
   181  011376                          150$:
   182  011376  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        011402  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        011404  000240                          NOP                     ;RETURN HERE IF ERROR
        011406  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        011410  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011412  000137  011710                  JMP     350$            ;GO TO 350$ IF ERROR
   183  011416                          160$:
   184  011416  004737  046060                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        011422  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
        011424  000240                          NOP                     ;RETURN HERE IF ERROR
        011426  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        011430  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011432  000137  011710                  JMP     350$            ;GO TO 350$ IF ERROR
   185  011436                          180$:
   186
   187                                  ;CHANGE LOOP ADDRESSES
   188  011436  012737  011446  001124          MOV     #190$,$LPERR
   189  011444  000410                          BR      200$                    ;SKIP TO NEXT OPERATION
   190
   191                                  ;;*****************************************************************************
   192                                  ;LOOP #3        READ
   193
   194  011446                          190$:
   195
   196                                  ;PREPARE DEVICE FOR TEST
   197  011446  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        011452  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        011454  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
        011456  000240                          NOP                     ;RETURN HERE IF ERROR
        011460  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        011462  000137  011710                  JMP     350$            ;GO TO 350$ IF ERROR
   198  011466                          200$:
   199
   200                                  ;SETUP PARAMETERS AND EXCUTE COMMAND
   201  011466                          240$:
   202  011466  012737  107324  001414          MOV     #BUFTWO+4,RMBAO         ;CHANGE MEMORY ADDRESS
   203  011474  012737  000071  001410          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
   204  011502  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
   205  011506  112722  000006                  MOVB    #RMDA,(R2)+
   206  011512  112722  000032                  MOVB    #RMOF,(R2)+
   207  011516  112722  000034                  MOVB    #RMDC,(R2)+
   208  011522  112722  000004                  MOVB    #RMBA,(R2)+
   209  011526  112722  000002                  MOVB    #RMWC,(R2)+
   210  011532  112722  000000                  MOVB    #RMCS1,(R2)+
   211  011536  112712  000200                  MOVB    #200,(R2)
```

```
                                ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011542  004737  044500          JSR     PC,PUT
011546  000404                  BR      250$            ;GO TO 250$ IF NO ERROR
011550  000240                  NOP                     ;RETURN HERE IF ERROR
011552  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
011554  000137  011710          JMP     350$            ;GO TO 350$ IF ERROR
011560                  250$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
011560  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                ;WAIT FOR COMMAND TO COMPLETE
011564  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                ;GO GET REGISTER STATUS
011570  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
011574  000404                  BR      260$            ;GO TO 260$ IF NO ERROR
011576  000240                  NOP                     ;RETURN HERE IF ERROR
011600  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
011602  000137  011710          JMP     350$            ;GO TO 350$ IF ERROR
011606                  260$:

                                ;VERIFY RESULTS OF READ COMMAND
011606  004737  045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
011612  000405                  BR      270$            ;GO TO 270$ IF NO ERROR
011614  000240                  NOP                     ;RETURN HERE IF ERROR
011616  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
011620  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011622  000137  011710          JMP     350$            ;GO TO 350$ IF ERROR
011626                  270$:
011626  004737  057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
011632  000405                  BR      280$            ;GO TO 280$ IF NO ERROR
011634  000240                  NOP                     ;RETURN HERE IF ERROR
011636  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
011640  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011642  000137  011710          JMP     350$            ;GO TO 350$ IF ERROR
011646                  280$:
011646  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
011652  000405                  BR      300$            ;GO TO 300$ IF NO ERROR
011654  000240                  NOP                     ;RETURN HERE IF ERROR
011656  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
011660  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011662  000137  011710          JMP     350$            ;GO TO 350$ IF ERROR
011666                  300$:

                                ;GO VERIFY DATA
011666  004737  043602          JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
011672  106320                  .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
011674  107324                  .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
011676  000404                  BR      310$            ;GO TO 310$ IF NO ERROR
011700  000240                  NOP                     ;RETURN HERE IF ERROR
011702  104000                  EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
011704  000137  011710          JMP     350$            ;GO TO 350$ IF ERROR
011710                  310$:

011710                  350$:
```

```
                        ;;*****************************************************************
                        ;*TEST 5         WRITE, WRITE CHECK ZEROS
                        ;;*****************************************************************
011710                  TST5:
011710  000004                    SCOPE                   ;SCOPE CALL
011712  000240                    NOP                     ;START OF TEST
011714  012706  001100            MOV     #STACK,SP       ;INITIALIZE STACK POINTER
011720  013700  001276            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
011724  013701  001464            MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
011730  012737  000005  001226    MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

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

011736                  10$:

                        ;PREPARE DEVICE FOR TEST
011736  004737  040472            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
011742  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
011744  000404                    BR      20$             ;GO TO 20$ IF NO ERROR
011746  000240                    NOP                     ;RETURN HERE IF ERROR
011750  104000                    EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
011752  000137  012666            JMP     350$            ;GO TO 350$ IF ERROR

                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
011756                  20$:
011756  012737  000000  001444    MOV     #0,RMDC0                ;CYLINDER = 0
011764  012737  000000  001416    MOV     #0,RMDA0                ;TRACK = 0, SECTOR = 0
011772  012737  106314  001414    MOV     #BUFONE,RMBA0           ;BUS ADDRESS
012000  012737  177376  001412    MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
012006  012737  010000  001442    MOV     #FMT16,RMOF0            ;16 BIT FORMAT
012014  012737  000063  001410    MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND

                        ;VERIFY THAT SECTOR IS NOT BAD
012022  004737  041416            JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
012026  000405                    BR      25$             ;GO TO 25$ IF NO ERROR
012030  104401  070230            TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
012034  104000                    EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
012036  000137  012666            JMP     350$            ;GO TO 350$ IF ERROR
012042                  25$:
012042  012737  071730  001174    MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
012050  012737  000001  001176    MOV     #1,$TMP1                ;RANGE OF PATTERN
012056  004737  043344            JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                        ;SETUP PARAMETERS AND EXCUTE COMMAND
012062  012702  001551            MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
012066  112722  000034            MOVB    #RMDC,(R2)+
012072  112722  000006            MOVB    #RMDA,(R2)+
012076  112722  000004            MOVB    #RMBA,(R2)+
012102  112722  000002            MOVB    #RMWC,(R2)+
```

```
                                        MOVB    #RMOF,(R2)+
                                        MOVB    #RMCS1,(R2)+
                                        MOVB    #200,(R2)               ;TERMINATE TABLE
                                30$:
                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      40$             ;GO TO 40$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                40$:

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

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

                                ;GO GET REGISTER STATUS
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      50$             ;GO TO 50$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                50$:

                                ;VERIFY RESULTS OF WRITE COMMAND
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      60$             ;GO TO 60$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JMP     @(SP)+          ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                60$:

                                ;MOVE LOOP ADDRESSES TO NEXT OPERATION
                                        MOV     #70$,$LPERR
                                        BR      80$                     ;SKIP TO WRITE OPERATION

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

                                70$:

                                ;PREPARE DEVICE FOR TEST
                                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                                ;VERIFY RECALIBRATION
                                        BR      80$             ;GO TO 80$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
```

```
    299 012236                          80$:
    300
    301                                 ;SETUP PARAMETERS AND EXCUTE COMMAND
    302 012236                          120$:
    303 012236  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
    304 012244  012737  177400  001412          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
    305 012252  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
    306 012260  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    307 012264  112722  000006                  MOVB    #RMDA,(R2)+
    308 012270  112722  000034                  MOVB    #RMDC,(R2)+
    309 012274  112722  000032                  MOVB    #RMOF,(R2)+
    310 012300  112722  000004                  MOVB    #RMBA,(R2)+
    311 012304  112722  000002                  MOVB    #RMWC,(R2)+
    312 012310  112722  000000                  MOVB    #RMCS1,(R2)+
    313 012314  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
    314
    315 012320  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        012324  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        012326  000240                          NOP                     ;RETURN HERE IF ERROR
        012330  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        012332  000137  012666                  JMP     350$            ;GO TO 350$ IF ERROR
    316 012336                          130$:
    317
    318                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        012336  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    319
    320                                 ;WAIT FOR COMMAND TO COMPLETE
        012342  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    321
    322                                 ;GO GET REGISTER STATUS
    323 012346  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        012352  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        012354  000240                          NOP                     ;RETURN HERE IF ERROR
        012356  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        012360  000137  012666                  JMP     350$            ;GO TO 350$ IF ERROR
    324 012364                          140$:
    325
    326                                 ;VERIFY RESULTS OF WRITE COMMAND
    327 012364  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        012370  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        012372  000240                          NOP                     ;RETURN HERE IF ERROR
        012374  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        012376  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012400  000137  012666                  JMP     350$            ;GO TO 350$ IF ERROR
    328 012404                          150$:
    329 012404  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        012410  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        012412  000240                          NOP                     ;RETURN HERE IF ERROR
        012414  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        012416  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012420  000137  012666                  JMP     350$            ;GO TO 350$ IF ERROR
    330 012424                          160$:
    331 012424  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        012430  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
        012432  000240                          NOP                     ;RETURN HERE IF ERROR
        012434  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        012436  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
                012440  000137  012666                  JMP     350$            ;GO TO 350$ IF ERROR
    332 012444                                  180$:
    333
    334                                         ;CHANGE LOOP ADDRESSES
    335 012444  012737  012454  001124                    MOV     #190$,$LPERR
    336 012452  000410                                   BR      200$                    ;SKIP TO NEXT OPERATION
    337
    338                                         ;;*****************************************************************
    339                                         ;LOOP #3        WRITE CHECK DATA
    340
    341 012454                                  190$:
    342
    343                                         ;PREPARE DEVICE FOR TEST
    344 012454  004737  040472                           JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        012460  154130                                   .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                                ;VERIFY RECALIBRATION
        012462  000404                                   BR      200$            ;GO TO 200$ IF NO ERROR
        012464  000240                                   NOP                     ;RETURN HERE IF ERROR
        012466  104000                                   EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        012470  000137  012666                           JMP     350$            ;GO TO 350$ IF ERROR
    345 012474                                  200$:
    346
    347                                         ;SETUP PARAMETERS AND EXCUTE COMMAND
    348 012474                                  240$:
    349 012474  012737  000051  001410                   MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
    350 012502  012702  001551                           MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    351 012506  112722  000006                           MOVB    #RMDA,(R2)+
    352 012512  112722  000032                           MOVB    #RMOF,(R2)+
    353 012516  112722  000034                           MOVB    #RMDC,(R2)+
    354 012522  112722  000004                           MOVB    #RMBA,(R2)+
    355 012526  112722  000002                           MOVB    #RMWC,(R2)+
    356 012532  112722  000000                           MOVB    #RMCS1,(R2)+
    357 012536  112712  000200                           MOVB    #200,(R2)
    358
    359 012542  004737  044500                           JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        012546  000404                                   BR      250$            ;GO TO 250$ IF NO ERROR
        012550  000240                                   NOP                     ;RETURN HERE IF ERROR
        012552  104000                                   EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        012554  000137  012666                           JMP     350$            ;GO TO 350$ IF ERROR
    360 012560                                  250$:
    361
    362                                         ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        012560  004737  044144                           JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    363
    364                                         ;WAIT FOR COMMAND TO COMPLETE
        012564  004737  045042                           JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    365
    366                                         ;GO GET REGISTER STATUS
    367 012570  004737  044230                           JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        012574  000404                                   BR      260$            ;GO TO 260$ IF NO ERROR
        012576  000240                                   NOP                     ;RETURN HERE IF ERROR
```

```
                012600  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                012602  000137  012666          JMP     350$            ;GO TO 350$ IF ERROR
    368 012606                          260$:
    369
    370                                 ;VERIFY RESULTS OF WRITE CHECK COMMAND
    371 012606  004737  045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        012612  000405                  BR      270$            ;GO TO 270$ IF NO ERROR
        012614  000240                  NOP                     ;RETURN HERE IF ERROR
        012616  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        012620  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012622  000137  012666          JMP     350$            ;GO TO 350$ IF ERROR
    372 012626                          270$:
    373 012626  004737  057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        012632  000405                  BR      280$            ;GO TO 280$ IF NO ERROR
        012634  000240                  NOP                     ;RETURN HERE IF ERROR
        012636  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        012640  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012642  000137  012666          JMP     350$            ;GO TO 350$ IF ERROR
    374 012646                          280$:
    375 012646  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        012652  000405                  BR      300$            ;GO TO 300$ IF NO ERROR
        012654  000240                  NOP                     ;RETURN HERE IF ERROR
        012656  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        012660  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012662  000137  012666          JMP     350$            ;GO TO 350$ IF ERROR
    376 012666                          300$:
    377
    378 012666                          350$:
    379
    380                                 ;;*****************************************************************
                                        ;*TEST 6         WRITE, WRITE CHECK ZEROS W/ WCE ERROR
                                        ;;*****************************************************************
        012666                          TST6:
        012666  000004                  SCOPE                   ;SCOPE CALL
        012670  000240                  NOP                     ;START OF TEST
        012672  012706  001100          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        012676  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        012702  013701  001464          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        012706  012737  000006  001226  MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
    381
    382                                 ;;*****************************************************************
    383                                 ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
    384
    385 012714                          10$:
    386
    387                                 ;PREPARE DEVICE FOR TEST
    388 012714  004737  040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        012720  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
        012722  000404                  BR      20$             ;GO TO 20$ IF NO ERROR
        012724  000240                  NOP                     ;RETURN HERE IF ERROR
```

```
        012726  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        012730  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
   389
   390                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
   391 012734                           20$:
   392 012734  012737  000000  001444          MOV     #0,RMDCO                ;CYLINDER = 0
   393 012742  012737  000000  001416          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
   394 012750  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
   395 012756  012737  177376  001412          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
   396 012764  012737  010000  001442          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
   397 012772  012737  000063  001410          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
   398
   399                                  ;VERIFY THAT SECTOR IS NOT BAD
        013000  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        013004  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
        013006  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        013012  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        013014  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
   400 013020                           25$:
   401 013020  012737  071730  001174          MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
   402 013026  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
   403 013034  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
   404
   405 013040  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
   406 013044  112722  000034                  MOVB    #RMDC,(R2)+
   407 013050  112722  000006                  MOVB    #RMDA,(R2)+
   408 013054  112722  000004                  MOVB    #RMBA,(R2)+
   409 013060  112722  000002                  MOVB    #RMWC,(R2)+
   410 013064  112722  000032                  MOVB    #RMOF,(R2)+
   411 013070  112722  000000                  MOVB    #RMCS1,(R2)+
   412 013074  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
   413 013100                           30$:
   414 013100  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        013104  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        013106  000240                          NOP                     ;RETURN HERE IF ERROR
        013110  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        013112  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
   415 013116                           40$:
   416
   417                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        013116  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
   418
   419                                  ;WAIT FOR COMMAND TO COMPLETE
        013122  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
   420
   421                                  ;GO EGT REGISTER STATUS
   422 013126  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        013132  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        013134  000240                          NOP                     ;RETURN HERE IF ERROR
        013136  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        013140  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
   423 013144                           50$:
   424
   425                                  ;VERIFY RESULTS OF WRITE COMMAND
   426 013144  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        013150  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
        013152  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        013154  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        013156  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013160  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
    427 013164                          60$:
    428
    429                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
    430 013164  012737  013174  001124          MOV     #70$,$LPERR
    431 013172  000410                          BR      80$                     ;SKIP TO WRITE OPERATION
    432
    433                                 ;;*****************************************************************
    434                                 ;LOOP #2        WRITE,WRITE CHECK DATA
    435
    436 013174                          70$:
    437
    438                                 ;PREPARE DEVICE FOR TEST
    439 013174  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        013200  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        013202  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        013204  000240                          NOP                     ;RETURN HERE IF ERROR
        013206  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        013210  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
    440 013214                          80$:
    441
    442                                 ;SETUP PARAMETERS AND EXCUTE COMMAND
    443 013214                          120$:
    444 013214  012737  177400  001412          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
    445 013222  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
    446 013230  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
    447 013236  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    448 013242  112722  000006                  MOVB    #RMDA,(R2)+
    449 013246  112722  000034                  MOVB    #RMDC,(R2)+
    450 013252  112722  000032                  MOVB    #RMOF,(R2)+
    451 013256  112722  000004                  MOVB    #RMBA,(R2)+
    452 013262  112722  000002                  MOVB    #RMWC,(R2)+
    453 013266  112722  000000                  MOVB    #RMCS1,(R2)+
    454 013272  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
    455
    456 013276  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        013302  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        013304  000240                          NOP                     ;RETURN HERE IF ERROR
        013306  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        013310  000137  014062                  JMP     350$            ;GO TO 350$ IF ERROR
    457 013314                          130$:
    458
    459                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        013314  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    460
    461                                 ;WAIT FOR COMMAND TO COMPLETE
        013320  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    462
```

```
                                ;GO READ STATUS FOR WRITE COMMAND
013324  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
013330  000404                        BR      140$            ;GO TO 140$ IF NO ERROR
013332  000240                        NOP                     ;RETURN HERE IF ERROR
013334  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
013336  000137  014062                JMP     350$            ;GO TO 350$ IF ERROR
013342                          140$:

                                ;CHECK FOR ERRORS DURING WRITE OPERATION
013342  004737  045226                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
013346  000405                        BR      150$            ;GO TO 150$ IF NO ERROR
013350  000240                        NOP                     ;RETURN HERE IF ERROR
013352  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
013354  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013356  000137  014062                JMP     350$            ;GO TO 350$ IF ERROR
013362                          150$:
013362  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
013366  000405                        BR      160$            ;GO TO 160$ IF NO ERROR
013370  000240                        NOP                     ;RETURN HERE IF ERROR
013372  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
013374  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013376  000137  014062                JMP     350$            ;GO TO 350$ IF ERROR
013402                          160$:
013402  004737  046060                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
013406  000405                        BR      180$            ;GO TO 180$ IF NO ERROR
013410  000240                        NOP                     ;RETURN HERE IF ERROR
013412  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
013414  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013416  000137  014062                JMP     350$            ;GO TO 350$ IF ERROR
013422                          180$:

                                ;CHANGE LOOP ADDRESSES
013422  012737  013430  001124        MOV     #190$,$LPERR

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

013430                          190$:

013430  012703  000001                MOV     #1,R3                   ;R3+WCE BIT POSITION
013434  050337  107316          195$: BIS     R3,BUFTWO-2             ;CHANGE LAST WORD OF BUFFER

                                ;PREPARE DEVICE FOR TEST
013440  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
013444  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
013446  000404                        BR      200$            ;GO TO 200$ IF NO ERROR
013450  000240                        NOP                     ;RETURN HERE IF ERROR
013452  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
013454  000137  014062                JMP     350$            ;GO TO 350$ IF ERROR
013460                          200$:
```

```
                                        ;READ DATA FROM DEVICE
013460                                  240$:
013460  012737  000051  001410                MOV     #WCD!GO,RMCS1O          ;WRITE CHECK DATA DATA COMMAND
013466  012702  001551                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
013472  112722  000006                        MOVB    #RMDA,(R2)+
013476  112722  000032                        MOVB    #RMOF,(R2)+
013502  112722  000034                        MOVB    #RMDC,(R2)+
013506  112722  000004                        MOVB    #RMBA,(R2)+
013512  112722  000002                        MOVB    #RMWC,(R2)+
013516  112722  000000                        MOVB    #RMCS1,(R2)+
013522  112712  000200                        MOVB    #200,(R2)

013526  004737  044500                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
013532  000404                              BR      250$            ;GO TO 250$ IF NO ERROR
013534  000240                              NOP                     ;RETURN HERE IF ERROR
013536  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
013540  000137  014062                      JMP     350$            ;GO TO 350$ IF ERROR
013544'                                 250$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
013544  004737  044144                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
013550  004737  045042                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
013554  004737  044230                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
013560  000404                              BR      260$            ;GO TO 260$ IF NO ERROR
013562  000240                              NOP                     ;RETURN HERE IF ERROR
013564  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
013566  000137  014062                      JMP     350$            ;GO TO 350$ IF ERROR
013572                                  260$:

                                        ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
013572  004737  045226                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
013576  000405                              BR      270$            ;GO TO 270$ IF NO ERROR
013600  000240                              NOP                     ;RETURN HERE IF ERROR
013602  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
013604  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013606  000137  014062                      JMP     350$            ;GO TO 350$ IF ERROR
013612                                  270$:
013612  032737  040000  001344                BIT     #WCE,RMCS2I             ;WAS 'WCE' DETECTED??
013620  001030                              BNE     285$                    ;YES!!

013622  004737  057742                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
013626  000405                              BR      280$            ;GO TO 280$ IF NO ERROR
013630  000240                              NOP                     ;RETURN HERE IF ERROR
013632  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
013634  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013636  000137  014062                      JMP     350$            ;GO TO 350$ IF ERROR
013642                                  280$:
013642  013737  001344  001140                MOV     RMCS2I,$GDDAT           ;EXPECTED STATUS
013650  052737  040000  001140                BIS     #WCE,$GDDAT
013656  013737  001344  001142                MOV     RMCS2I,$BDDAT           ;RECEIVED STATUS
013664  010337  001174                        MOV     R3,$TMP0                ;FAILING BIT POSITION
013670  012737  107316  001176                MOV     #BUFTWO-2,$TMP1         ;FAILING ADDRESS
```

```
                                        EMT     337
                                        BR      350$
                                285$:
                                        MOVB    #RMDB,GETINX    ;SETUP GET INDEX TABLE
                                        MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
                                        JSR     PC,GET          ;GO READ RMDB VIA GET SUBROUTINE
                                        BR      290$            ;GO TO 290$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR DEFINED BY GET SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                290$:
                                        MOV     RMDBI,$BDDAT            ;RECEIVED DATA
                                        MOV     BUFTWO-2,$GDDAT         ;EXPECTED DATA
                                        BIC     R3,$GDDAT
                                        MOV     #BUFTWO-2,$GDADR        ;EXPECTED ADDRESS
                                        MOV     RMBAI,$BDADR            ;RECEIVED ADDRESS
                                        SUB     #2,$BDADR               ;ADJUST BUS ADDRESS
                                        CMP     $GDADR,$BDADR           ;ADDRESSES OK??
                                        BEQ     295$                    ;YES!!
                                        EMT     340
                                        BR      350$
                                295$:   CMP     $GDDAT,$BDDAT           ;DATA OK??
                                        BEQ     296$                    ;YES!!
                                        EMT     341
                                        BR      350$
                                296$:

                                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                                        BR      300$            ;GO TO 300$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                300$:
                                        BIC     R3,BUFTWO-2             ;RESTORE DATA PATTERN
                                        ASL     R3                      ;SHIFT TO NEXT BIT
                                        BEQ     350$                    ;EXIT IF DONE
                                        JMP     195$                    ;REPEAT TEST FOR NEXT DATA BIT
                                350$:

                                ;;**************************************************************************
                                ;*TEST 7         WRITE, READ ONES
                                ;;**************************************************************************
                                TST7:
                                        SCOPE                   ;SCOPE CALL
                                        NOP                     ;START OF TEST
                                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                                        MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

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

                                10$:

                                ;PREPARE THE DEVICE FOR TEST
```

```
563 014110  004737  040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    014114  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
    014116  000404                  BR      20$             ;GO TO 20$ IF NO ERROR
    014120  000240                  NOP                     ;RETURN HERE IF ERROR
    014122  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    014124  000137  015070          JMP     350$            ;GO TO 350$ IF ERROR
564
565                             ;LOAD PARAMETERS AND GENERATE DATA BUFFER
566 014130                      20$:
567 014130  012737  000000  001444          MOV     #0,RMDCO                ;CYLINDER = 0
568 014136  012737  000000  001416          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
569 014144  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
570 014152  012737  177376  001412          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
571 014160  012737  010000  001442          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
572 014166  012737  000063  001410          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
573
574                             ;VERIFY THAT SECTOR IS NOT BAD
    014174  004737  041416          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    014200  000405                  BR      25$             ;GO TO 25$ IF NO ERROR
    014202  104401  070230          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    014206  104000                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    014210  000137  015070          JMP     350$            ;GO TO 350$ IF ERROR
575 014214                      25$:
576 014214  012737  071666  001174          MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
577 014222  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
578 014230  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
579
580                             ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
581 014234  012702  001551          MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
582 014240  112722  000034          MOVB    #RMDC,(R2)+
583 014244  112722  000006          MOVB    #RMDA,(R2)+
584 014250  112722  000004          MOVB    #RMBA,(R2)+
585 014254  112722  000002          MOVB    #RMWC,(R2)+
586 014260  112722  000032          MOVB    #RMOF,(R2)+
587 014264  112722  000000          MOVB    #RMCS1,(R2)+
588 014270  112712  000200          MOVB    #200,(R2)               ;TERMINATE TABLE
589 014274                      30$:
590
591                             ;FORMAT THE DRIVE
592 014274  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    014300  000404                  BR      40$             ;GO TO 40$ IF NO ERROR
    014302  000240                  NOP                     ;RETURN HERE IF ERROR
    014304  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    014306  000137  015070          JMP     350$            ;GO TO 350$ IF ERROR
593 014312                      40$:
594
595                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    014312  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
596
597                             ;WAIT FOR COMMAND TO COMPLETE
```

```
      014316  004737  045042                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  598
  599                                   ;GO READ STATUS FOR FORMAT OPERATION
  600 014322  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      014326  000404                        BR      50$             ;GO TO 50$ IF NO ERROR
      014330  000240                        NOP                     ;RETURN HERE IF ERROR
      014332  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      014334  000137  015070                JMP     350$            ;GO TO 350$ IF ERROR
  601 014340                            50$:
  602
  603                                   ;VERIFY NO ERRORS DURING FORMAT
  604 014340  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      014344  000405                        BR      60$             ;GO TO 60$ IF NO ERROR
      014346  000240                        NOP                     ;RETURN HERE IF ERROR
      014350  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      014352  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      014354  000137  015070                JMP     350$            ;GO TO 350$ IF ERROR
  605 014360                            60$:
  606
  607                                   ;MOVE LOOP ADDRESSES TO NEXT OPERATION
  608 014360  012737  014370  001124        MOV     #70$,$LPERR
  609 014366  000410                        BR      80$                     ;SKIP TO WRITE OPERATION
  610
  611                                   ;;*************************************************************************
  612                                   ;LOOP #2        WRITE,READ
  613
  614 014370                            70$:
  615
  616
  617                                   ;PREPARE DEVICE FOR TEST
  618 014370  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      014374  154130                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                    ;VERIFY RECALIBRATION
      014376  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
      014400  000240                        NOP                     ;RETURN HERE IF ERROR
      014402  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      014404  000137  015070                JMP     350$            ;GO TO 350$ IF ERROR
  619 014410                            80$:
  620
  621                                   ;WRITE DATA TO THE DRIVE
  622 014410                            120$:
  623 014410  012737  106320  001414        MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
  624 014416  012737  177400  001412        MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
  625 014424  012737  000061  001410        MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
  626 014432  012702  001551                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  627 014436  112722  000006                MOVB    #RMDA,(R2)+
  628 014442  112722  000034                MOVB    #RMDC,(R2)+
  629 014446  112722  000032                MOVB    #RMOF,(R2)+
  630 014452  112722  000004                MOVB    #RMBA,(R2)+
  631 014456  112722  000002                MOVB    #RMWC,(R2)+
  632 014462  112722  000000                MOVB    #RMCS1,(R2)+
```

```
                                        MOVB    #200,(R2)+              ;TERMINATE TABLE
                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      130$            ;GO TO 130$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                130$:

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

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

                                ;GO READ STATUS FOR WRITE COMMAND
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      140$            ;GO TO 140$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                140$:

                                ;CHECK FOR ERRORS DURING WRITE OPERATION
                                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                                        BR      150$            ;GO TO 150$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                150$:
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      160$            ;GO TO 160$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                160$:
                                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                                        BR      180$            ;GO TO 180$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                180$:

                                ;CHANGE LOOP ADDRESSES
                                        MOV     #190$,$LPERR
                                        BR      200$                    ;SKIP TO NEXT OPERATION

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

                                190$:

                                ;PREPARE DEVICE FOR TEST
                                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
```

```
        014632  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
        014634  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
        014636  000240                          NOP                     ;RETURN HERE IF ERROR
        014640  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        014642  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
    665 014646                          200$:
    666
    667                                 ;READ DATA FROM DEVICE
    668 014646                          240$:
    669 014646  012737  107324  001414          MOV     #BUFTWO+4,RMBA0         ;CHANGE MEMORY ADDRESS
    670 014654  012737  000071  001410          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
    671 014662  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    672 014666  112722  000006                  MOVB    #RMDA,(R2)+
    673 014672  112722  000032                  MOVB    #RMOF,(R2)+
    674 014676  112722  000034                  MOVB    #RMDC,(R2)+
    675 014702  112722  000004                  MOVB    #RMBA,(R2)+
    676 014706  112722  000002                  MOVB    #RMWC,(R2)+
    677 014712  112722  000000                  MOVB    #RMCS1,(R2)+
    678 014716  112712  000200                  MOVB    #200,(R2)
    679
    680 014722  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        014726  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
        014730  000240                          NOP                     ;RETURN HERE IF ERROR
        014732  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        014734  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
    681 014740                          250$:
    682
    683                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        014740  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    684
    685                                 ;WAIT FOR COMMAND TO COMPLETE
        014744  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    686
    687                                 ;GO READ STATUS FOR READ OPERATION
    688 014750  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        014754  000404                          BR      260$            ;GO TO 260$ IF NO ERROR
        014756  000240                          NOP                     ;RETURN HERE IF ERROR
        014760  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014762  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
    689 014766                          260$:
    690
    691                                 ;CHECK FOR ERRORS DURING READ OPERATION
    692 014766  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        014772  000405                          BR      270$            ;GO TO 270$ IF NO ERROR
        014774  000240                          NOP                     ;RETURN HERE IF ERROR
        014776  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        015000  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015002  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
    693 015006                          270$:
    694 015006  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
```

```
        015012  000405                          BR      280$            ;GO TO 280$ IF NO ERROR
        015014  000240                          NOP                     ;RETURN HERE IF ERROR
        015016  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        015020  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015022  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
   695  015026                          280$:
   696  015026  004737  046060                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        015032  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
        015034  000240                          NOP                     ;RETURN HERE IF ERROR
        015036  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        015040  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015042  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
   697  015046                          300$:
   698  015046  004737  043602                JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        015052  106320                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        015054  107324                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        015056  000404                          BR      310$            ;GO TO 310$ IF NO ERROR
        015060  000240                          NOP                     ;RETURN HERE IF ERROR
        015062  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        015064  000137  015070                  JMP     350$            ;GO TO 350$ IF ERROR
   699  015070                          310$:
   700
   701  015070                          350$:
   702
   703                                  ;;*****************************************************************
                                        ;*TEST 10        WRITE, WRITE CHECK ONES
                                        ;;*****************************************************************
        015070                          TST10:
        015070  000004                          SCOPE                   ;SCOPE CALL
        015072  000240                          NOP                     ;START OF TEST
        015074  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        015100  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        015104  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        015110  012737  000010  001226          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   704
   705                                  ;;*****************************************************************
   706                                  ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
   707
   708  015116                          10$:
   709
   710                                  ;PREPARE THE DEVICE FOR TEST
   711  015116  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        015122  154130                          .WORD   154130          ;'ASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        015124  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        015126  000240                          NOP                     ;RETURN HERE IF ERROR
        015130  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        015132  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
   712
   713                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
   714  015136                          20$:
```

```
    715 015136  012737  000000  001444          MOV     #0,RMDCO                ;CYLINDER = 0
    716 015144  012737  000000  001416          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
    717 015152  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
    718 015160  012737  177376  001412          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
    719 015166  012737  010000  001442          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
    720 015174  012737  000063  001410          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
    721
    722                                 ;VERIFY THAT SECTOR IS NOT BAD
        015202  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        015206  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
        015210  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        015214  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        015216  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    723 015222                          25$:
    724 015222  012737  071666  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
    725 015230  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
    726 015236  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
    727
    728                                 ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
    729 015242  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
    730 015246  112722  000034                  MOVB    #RMDC,(R2)+
    731 015252  112722  000006                  MOVB    #RMDA,(R2)+
    732 015256  112722  000004                  MOVB    #RMBA,(R2)+
    733 015262  112722  000002                  MOVB    #RMWC,(R2)+
    734 015266  112722  000032                  MOVB    #RMOF,(R2)+
    735 015272  112722  000000                  MOVB    #RMCS1,(R2)+
    736 015276  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
    737 015302                          30$:
    738
    739                                 ;FORMAT THE DRIVE
    740 015302  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        015306  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        015310  000240                          NOP                     ;RETURN HERE IF ERROR
        015312  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        015314  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    741 015320                          40$:
    742
    743                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        015320  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    744
    745                                 ;WAIT FOR COMMAND TO COMPLETE
        015324  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    746
    747                                 ;GO READ STATUS FOR FORMAT OPERATION
    748 015330  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015334  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        015336  000240                          NOP                     ;RETURN HERE IF ERROR
        015340  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015342  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    749 015346                          50$:
    750
    751                                 ;VERIFY NO ERRORS DURING FORMAT
    752 015346  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        015352  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
        015354  000240                          NOP                     ;RETURN HERE IF ERROR
        015356  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        015360  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
                                                                    D 8
015362  000137  016046                    JMP     350$            ;GO TO 350$ IF ERROR
753 015366                          60$:
754
755                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
756 015366  012737  015376  001124          MOV     #70$,$LPERR
757 015374  000410                          BR      80$                     ;SKIP TO WRITE OPERATION
758
759                                 ;;*****************************************************************
760                                 ;LOOP #2        WRITE,WRITE CHECK DATA
761
762 015376                          70$:
763
764
765                                 ;PREPARE DEVICE FOR WRITE OPERATION
766 015376  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    015402  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                    ;VERIFY RECALIBRATION
    015404  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
    015406  000240                          NOP                     ;RETURN HERE IF ERROR
    015410  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    015412  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
767 015416                          80$:
768
769                                 ;WRITE DATA TO THE DRIVE
770 015416                          120$:
771 015416  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
772 015424  012737  177400  001412          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
773 015432  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
774 015440  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
775 015444  112722  000006                  MOVB    #RMDA,(R2)+
776 015450  112722  000034                  MOVB    #RMDC,(R2)+
777 015454  112722  000032                  MOVB    #RMOF,(R2)+
778 015460  112722  000004                  MOVB    #RMBA,(R2)+
779 015464  112722  000002                  MOVB    #RMWC,(R2)+
780 015470  112722  000000                  MOVB    #RMCS1,(R2)+
781 015474  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
782
783 015500  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    015504  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
    015506  000240                          NOP                     ;RETURN HERE IF ERROR
    015510  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    015512  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
784 015516                          130$:
785
786                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    015516  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
787
788                                 ;WAIT FOR COMMAND TO COMPLETE
    015522  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
789
790                                 ;GO READ STATUS FOR WRITE COMMAND
```

```
    791 015526  004737  044230              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015532  000404                      BR      140$            ;GO TO 140$ IF NO ERROR
        015534  000240                      NOP                     ;RETURN HERE IF ERROR
        015536  104000                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015540  000137  016046              JMP     350$            ;GO TO 350$ IF ERROR
    792 015544                          140$:
    793
    794
    795 015544  004737  045226          ;CHECK FOR ERRORS DURING WRITE OPERATION
                                            JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        015550  000405                      BR      150$            ;GO TO 150$ IF NO ERROR
        015552  000240                      NOP                     ;RETURN HERE IF ERROR
        015554  104000                      EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        015556  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015560  000137  016046              JMP     350$            ;GO TO 350$ IF ERROR
    796 015564                          150$:
    797 015564  004737  057742              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        015570  000405                      BR      160$            ;GO TO 160$ IF NO ERROR
        015572  000240                      NOP                     ;RETURN HERE IF ERROR
        015574  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        015576  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015600  000137  016046              JMP     350$            ;GO TO 350$ IF ERROR
    798 015604                          160$:
    799 015604  004737  046060              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        015610  000405                      BR      180$            ;GO TO 180$ IF NO ERROR
        015612  000240                      NOP                     ;RETURN HERE IF ERROR
        015614  104000                      EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        015616  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015620  000137  016046              JMP     350$            ;GO TO 350$ IF ERROR
    800 015624                          180$:
    801
    802                                 ;CHANGE LOOP ADDRESSES
    803 015624  012737  015634  001124          MOV     #190$,$LPERR
    804 015632  000410                        BR      200$                    ;SKIP TO NEXT OPERATION
    805
    806                                 ;;*****************************************************************
    807                                 ;LOOP #3        WRITE CHECK DATA
    808
    809 015634                          190$:
    810
    811                                 ;PREPARE DEVICE FOR WRITE CHECK DATA OPERATION
    812 015634  004737  040472              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        015640  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                    ;VERIFY RECALIBRATION
        015642  000404                      BR      200$            ;GO TO 200$ IF NO ERROR
        015644  000240                      NOP                     ;RETURN HERE IF ERROR
        015646  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        015650  000137  016046              JMP     350$            ;GO TO 350$ IF ERROR
    813 015654                          200$:
    814
    815                                 ;WRITE CHECK DATA DATA FROM DEVICE
    816 015654                          240$:
```

```
    817 015654  012737  000051  001410          MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
    818 015662  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    819 015666  112722  000006                  MOVB    #RMDA,(R2)+
    820 015672  112722  000032                  MOVB    #RMOF,(R2)+
    821 015676  112722  000034                  MOVB    #RMDC,(R2)+
    822 015702  112722  000004                  MOVB    #RMBA,(R2)+
    823 015706  112722  000002                  MOVB    #RMWC,(R2)+
    824 015712  112722  000000                  MOVB    #RMCS1,(R2)+
    825 015716  112712  000200                  MOVB    #200,(R2)
    826
    827 015722  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        015726  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
        015730  000240                          NOP                     ;RETURN HERE IF ERROR
        015732  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        015734  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    828 015740                          250$:
    829
    830                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        015740  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    831
    832                                 ;WAIT FOR COMMAND TO COMPLETE
        015744  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    833
    834                                 ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
    835 015750  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015754  000404                          BR      260$            ;GO TO 260$ IF NO ERROR
        015756  000240                          NOP                     ;RETURN HERE IF ERROR
        015760  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015762  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    836 015766                          260$:
    837
    838                                 ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
    839 015766  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        015772  000405                          BR      270$            ;GO TO 270$ IF NO ERROR
        015774  000240                          NOP                     ;RETURN HERE IF ERROR
        015776  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        016000  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016002  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    840 016006                          270$:
    841 016006  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        016012  000405                          BR      280$            ;GO TO 280$ IF NO ERROR
        016014  000240                          NOP                     ;RETURN HERE IF ERROR
        016016  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        016020  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016022  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    842 016026                          280$:
    843 016026  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        016032  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
        016034  000240                          NOP                     ;RETURN HERE IF ERROR
        016036  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        016040  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016042  000137  016046                  JMP     350$            ;GO TO 350$ IF ERROR
    844 016046                          300$:
    845
    846 016046                          350$:
    847
    848                                 ;;*****************************************************************
```

```
                                        ;*TEST 11       WRITE, WRITE CHECK ONES W/ WCE ERROR
                                        ;;*****************************************************************
016046                                  TST11:
016046  000004                                  SCOPE                   ;SCOPE CALL
016050  000240                                  NOP                     ;START OF TEST
016052  012706  001100                          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
016056  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
016062  013701  001464                          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
016066  012737  000011  001226                  MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
849
850                                     ;;*****************************************************************
851                                     ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
852
853 016074                              10$:
854
855                                     ;PREPARE THE DEVICE FOR FORMAT OPERATION
856 016074  004737  040472                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    016100  154130                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
    016102  000404                              BR      20$             ;GO TO 20$ IF NO ERROR
    016104  000240                              NOP                     ;RETURN HERE IF ERROR
    016106  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    016110  000137  017244                      JMP     350$            ;GO TO 350$ IF ERROR
857 016114                              20$:
858
859                                     ;LOAD PARAMETERS AND GENERATE DATA BUFFER
860 016114  012737  000000  001444              MOV     #0,RMDCO                ;CYLINDER = 0
861 016122  012737  000000  001416              MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
862 016130  012737  106314  001414              MOV     #BUFONE,RMBAO           ;BUS ADDRESS
863 016136  012737  177376  001412              MOV     #-258.,RMWCO            ;WORD COUNT = 1 SECTOR
864 016144  012737  010000  001442              MOV     #FMT16,RMOFO            ;16 BIT FORMAT
865 016152  012737  000063  001410              MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
866
867                                     ;VERIFY THAT SECTOR IS NOT BAD
    016160  004737  041416                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    016164  000405                              BR      25$             ;GO TO 25$ IF NO ERROR
    016166  104401  070230                      TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    016172  104000                              EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    016174  000137  017244                      JMP     350$            ;GO TO 350$ IF ERROR
868 016200                              25$:
869 016200  012737  071666  001174              MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
870 016206  012737  000001  001176              MOV     #1,$TMP1                ;RANGE OF PATTERN
871 016214  004737  043344                      JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
872
873                                     ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
874 016220  012702  001551              MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
875 016224  112722  000034              MOVB    #RMDC,(R2)+
876 016230  112722  000006              MOVB    #RMDA,(R2)+
877 016234  112722  000004              MOVB    #RMBA,(R2)+
878 016240  112722  000002              MOVB    #RMWC,(R2)+
879 016244  112722  000032              MOVB    #RMOF,(R2)+
```

```
    880 016250  112722  000000                  MOVB    #RMCS1,(R2)+
    881 016254  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
    882 016260                          30$:
    883
    884                                 ;FORMAT THE DRIVE
    885 016260  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        016264  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        016266  000240                          NOP                     ;RETURN HERE IF ERROR
        016270  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        016272  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    886 016276                          40$:
    887
    888                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        016276  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    889
    890                                 ;WAIT FOR COMMAND TO COMPLETE
        016302  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    891
    892                                 ;GO READ STATUS FOR FORMAT OPERATION
    893 016306  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        016312  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        016314  000240                          NOP                     ;RETURN HERE IF ERROR
        016316  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        016320  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    894 016324                          50$:
    895
    896                                 ;VERIFY NO ERRORS DURING FORMAT
    897 016324  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        016330  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
        016332  000240                          NOP                     ;RETURN HERE IF ERROR
        016334  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        016336  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016340  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    898 016344                          60$:
    899
    900                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
    901 016344  012737  016354  001124          MOV     #70$,$LPERR
    902 016352  000410                          BR      80$                     ;SKIP TO WRITE OPERATION
    903
    904                                 ;;*****************************************************************
    905                                 ;LOOP #2        WRITE,WRITE CHECK DATA
    906
    907 016354                          70$:
    908
    909                                 ;PREPARE DEVICE FOR WRITE OPERATION
    910 016354  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        016360  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        016362  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        016364  000240                          NOP                     ;RETURN HERE IF ERROR
        016366  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
```

```
        016370  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    911 016374                          80$:
    912
    913                                 ;WRITE DATA TO THE DRIVE
    914 016374                          120$:
    915 016374  012737  177400  001412          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
    916 016402  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
    917 016410  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
    918 016416  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    919 016422  112722  000006                  MOVB    #RMDA,(R2)+
    920 016426  112722  000034                  MOVB    #RMDC,(R2)+
    921 016432  112722  000032                  MOVB    #RMOF,(R2)+
    922 016436  112722  000004                  MOVB    #RMBA,(R2)+
    923 016442  112722  000002                  MOVB    #RMWC,(R2)+
    924 016446  112722  000000                  MOVB    #RMCS1,(R2)+
    925 016452  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
    926
    927 016456  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        016462  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        016464  000240                          NOP                     ;RETURN HERE IF ERROR
        016466  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        016470  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    928 016474                          130$:
    929
    930                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        016474  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    931
    932                                 ;WAIT FOR COMMAND TO COMPLETE
        016500  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    933
    934                                 ;GO READ STATUS FOR WRITE COMMAND
    935 016504  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        016510  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        016512  000240                          NOP                     ;RETURN HERE IF ERROR
        016514  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        016516  000137  017244                  JMP     350$            ;GO TO 350$ IF ERRUR
    936 016522                          140$:
    937
    938                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
    939 016522  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        016526  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        016530  000240                          NOP                     ;RETURN HERE IF ERROR
        016532  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        016534  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016536  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    940 016542                          150$:
    941 016542  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        016546  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        016550  000240                          NOP                     ;RETURN HERE IF ERROR
        016552  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        016554  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016556  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    942 016562                          160$:
    943 016562  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        016566  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
        016570  000240                          NOP                     ;RETURN HERE IF ERROR
        016572  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
```

```
        016574  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016576  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    944 016602                          180$:
    945
    946                                 ;CHANGE LOOP ADDRESSES
    947 016602  012737  016610  001124          MOV     #190$,$LPERR
    948
    949                                 ;;*****************************************************************
    950                                 ;LOOP #3        WRITE CHECK DATA
    951
    952 016610                          190$:
    953
    954 016610  012703  000001                  MOV     #1,R3                   ;R3+WCE BIT POSITION
    955 016614  040337  107316          195$:   BIC     R3,BUFTWO-2             ;CHANGE LAST WORD OF BUFFER
    956
    957                                 ;PREPARE DEVICE FOR WRITE CHECK DATA OPERATION
    958 016620  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        016624  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILARLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        016626  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
        016630  000240                          NOP                     ;RETURN HERE IF ERROR
        016632  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        016634  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    959 016640                          200$:
    960
    961                                 ;READ DATA FROM DEVICE
    962 016640                          240$:
    963 016640  012737  000051  001410          MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
    964 016646  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    965 016652  112722  000006                  MOVB    #RMDA,(R2)+
    966 016656  112722  000032                  MOVB    #RMOF,(R2)+
    967 016662  112722  000034                  MOVB    #RMDC,(R2)+
    968 016666  112722  000004                  MOVB    #RMBA,(R2)+
    969 016672  112722  000002                  MOVB    #RMWC,(R2)+
    970 016676  112722  000000                  MOVB    #RMCS1,(R2)+
    971 016702  112712  000200                  MOVB    #200,(R2)
    972
    973 016706  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        016712  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
        016714  000240                          NOP                     ;RETURN HERE IF ERROR
        016716  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        016720  000137  017244                  JMP     350$            ;GO TO 350$ IF ERROR
    974 016724                          250$:
    975
    976                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        016724  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    977
    978                                 ;WAIT FOR COMMAND TO COMPLETE
        016730  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    979
    980                                 ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
```

```
  981 016734  004737  044230                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      016740  000404                            BR      260$            ;GO TO 260$ IF NO ERROR
      016742  000240                            NOP                     ;RETURN HERE IF ERROR
      016744  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      016746  000137  017244                    JMP     350$            ;GO TO 350$ IF ERROR
  982 016752                              260$:
  983
  984                                         ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
  985 016752  004737  045226                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      016756  000405                            BR      270$            ;GO TO 270$ IF NO ERROR
      016760  000240                            NOP                     ;RETURN HERE IF ERROR
      016762  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      016764  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      016766  000137  017244                    JMP     350$            ;GO TO 350$ IF ERROR
  986 016772                              270$:
  987 016772  032737  040000  001344            BIT     #WCE,RMCS2I             ;WAS 'WCE' DETECTED??
  988 017000  001030                            BNE     285$                    ;YES!!
  989 017002  004737  057742                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      017006  000405                            BR      280$            ;GO TO 280$ IF NO ERROR
      017010  000240                            NOP                     ;RETURN HERE IF ERROR
      017012  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      017014  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      017016  000137  017244                    JMP     350$            ;GO TO 350$ IF ERROR
  990 017022                              280$:
  991 017022  013737  001344  001140            MOV     RMCS2I,$GDDAT           ;EXPECTED STATUS
  992 017030  052737  040000  001140            BIS     #WCE,$GDDAT
  993 017036  013737  001344  001142            MOV     RMCS2I,$BDDAT           ;RECEIVED STATUS
  994 017044  010337  001174                    MOV     R3,$TMP0                ;FAILING BIT POSITION
  995 017050  012737  107316  001176            MOV     #BUFTWO-2,$TMP1         ;FAILING ADDRESS
  996 017056  104337                            EMT     337
  997 017060  000471                            BR      350$
  998
  999 017062                              285$:
 1000 017062  112737  000022  001522            MOVB    #RMDB,GETINX    ;SETUP GET INDEX TABLE
      017070  112737  000200  001523            MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
      017076  012737  017244  001356            MOV     #350$,RMDBI     ;SET RMDB INPUT BUFFER = 350$
      017104  004737  044230                    JSR     PC,GET          ;GO READ RMDB VIA GET SUBROUTINE
      017110  000402                            BR      290$            ;GO TO 290$ IF NO ERROR
      017112  000240                            NOP                     ;RETURN HERE IF ERROR
      017114  104000                            EMT                     ;ERROR DEFINED BY GET SUBROUTINE
 1001
 1002 017116                              290$:
 1003 017116  013737  001356  001142            MOV     RMDBI,$BDDAT            ;RECEIVED DATA
 1004 017124  013737  107316  001140            MOV     BUFTWO-2,$GDDAT         ;EXPECTED DATA
 1005 017132  050337  001140                    BIS     R3,$GDDAT
 1006 017136  012737  107316  001134            MOV     #BUFTWO-2,$GDADR        ;EXPECTED ADDRESS
 1007 017144  013737  001340  001136            MOV     RMBAI,$BDADR            ;RECEIVED ADDRESS
 1008 017152  162737  000002  001136            SUB     #2,$BDADR               ;CORRECT MEMORY ADDRESS
 1009 017160  023737  001134  001136            CMP     $GDADR,$BDADR           ;ADDRESSES OK??
 1010 017166  001402                            BEQ     295$                    ;YES!!
 1011 017170  104340                            EMT     340
 1012 017172  000424                            BR      350$
 1013 017174  023737  001140  001142  295$:     CMP     $GDDAT,$BDDAT           ;DATA OK??
 1014 017202  001402                            BEQ     296$                    ;YES!!
 1015 017204  104341                            EMT     341
 1016 017206  000416                            BR      350$
 1017 017210                              296$:
```

```
1018
1019 017210  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     017214  000405                  BR      300$            ;GO TO 300$ IF NO ERROR
     017216  000240                  NOP                     ;RETURN HERE IF ERROR
     017220  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     017222  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017224  000137  017244          JMP     350$            ;GO TO 350$ IF ERROR
1020 017230                      300$:
1021 017230  050337  107316          BIS     R3,BUFTWO-2             ;RESTORE DATA PATTERN
1022 017234  006303                  ASL     R3                      ;SHIFT TO NEXT BIT
1023 017236  001402                  BEQ     350$                    ;EXIT IF DONE
1024 017240  000137  016614          JMP     195$                    ;REPEAT TEST FOR NEXT DATA BIT
1025 017244                      350$:
1026
1027                             ;;*******************************************************************
                                 ;*TEST 12        WRITE, WRITE CHECK MULTIPLE SECTORS
                                 ;;*******************************************************************
     017244                      TST12:
     017244  000004                  SCOPE                   ;SCOPE CALL
     017246  000240                  NOP                     ;START OF TEST
     017250  012706  001100          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     017254  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     017260  013701  001464          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     017264  012737  000012  001226  MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1028
1029                             ;;*******************************************************************
1030                             ;LOOP #1        FORMAT,WRITE,WRITE CHECK
1031
1032 017272                      10$:
1033
1034                             ;PREPARE THE DEVICE FOR FORMAT OPERATION
1035 017272  004737  040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     017276  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                             ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                             ;CLEAR CONTROLLER & SELECT DEVICE
                                                             ;VERIFY CONTROLLER CLEAR OPERATION
                                                             ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                             ;VERIFY PACK ACKNOWLEDGE
                                                             ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                             ;VERIFY RECALIBRATION
     017300  000404                  BR      20$             ;GO TO 20$ IF NO ERROR
     017302  000240                  NOP                     ;RETURN HERE IF ERROR
     017304  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     017306  000137  020250          JMP     350$            ;GO TO 350$ IF ERROR
1036 017312                      20$:
1037
1038                             ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1039 017312  012737  000000  001444      MOV     #0,RMDCO                ;CYLINDER = 0
1040 017320  012737  000000  001416      MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
1041 017326  012737  106314  001414      MOV     #BUFONE,RMBAO           ;BUS ADDRESS
1042 017334  012737  176774  001412      MOV     #-258.*2,RMWCO          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
1043 017342  012737  010000  001442      MOV     #FMT16,RMOFO            ;16 BIT FORMAT
1044 017350  012737  000063  001410      MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
1045
1046                             ;VERIFY THAT SECTOR IS NOT BAD
     017356  004737  041416          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     017362  000405                  BR      25$             ;GO TO 25$ IF NO ERROR
```

```
                017364  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
                017370  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
                017372  000137  020250                  JMP     350$            ;GO TO 350$ IF ERROR
     1047       017376                          25$:
     1048       017376  012737  071730  001174          MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
     1049       017404  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
     1050       017412  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
     1051
     1052                                          ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
     1053       017416  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
     1054       017422  112722  000034                  MOVB    #RMDC,(R2)+
     1055       017426  112722  000006                  MOVB    #RMDA,(R2)+
     1056       017432  112722  000004                  MOVB    #RMBA,(R2)+
     1057       017436  112722  000002                  MOVB    #RMWC,(R2)+
     1058       017442  112722  000032                  MOVB    #RMOF,(R2)+
     1059       017446  112722  000000                  MOVB    #RMCS1,(R2)+
     1060       017452  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
     1061       017456                          30$:
     1062
     1063                                          ;FORMAT THE DRIVE
     1064       017456  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                017462  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
                017464  000240                          NOP                     ;RETURN HERE IF ERROR
                017466  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                017470  000137  020250                  JMP     350$            ;GO TO 350$ IF ERROR
     1065       017474                          40$:
     1066
     1067                                          ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
                017474  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
     1068
     1069                                          ;WAIT FOR COMMAND TO COMPLETE
                017500  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
     1070
     1071                                          ;GO READ STATUS FOR FORMAT OPERATION
     1072       017504  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                017510  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
                017512  000240                          NOP                     ;RETURN HERE IF ERROR
                017514  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                017516  000137  020250                  JMP     350$            ;GO TO 350$ IF ERROR
     1073       017522                          50$:
     1074
     1075                                          ;VERIFY NO ERRORS DURING FORMAT
     1076       017522  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                017526  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
                017530  000240                          NOP                     ;RETURN HERE IF ERROR
                017532  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                017534  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017536  000137  020250                  JMP     350$            ;GO TO 350$ IF ERROR
     1077       017542                          60$:
     1078
     1079                                          ;MOVE LOOP ADDRESSES TO NEXT OPERATION
     1080       017542  012737  017614  001124          MOV     #70$,$LPERR
     1081
     1082                                          ;REGENERATE DATA BUFFER
     1083       017550  012737  177000  001412          MOV     #-256.*2,RMWCO  ;CHANGE WORD COUNT
     1084       017556  012737  106314  001414          MOV     #BUFONE,RMBAO           ;CHANGE BUS ADDRESS
     1085       017564  012737  000060  001410          MOV     #WD,RMCS10              ;WRITE DATA
```

```
 1086 017572  012737  071666  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
 1087 017600  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
 1088 017606  004737  043344                  JSR     PC,GENBUF
 1089 017612  000410                          BR      80$                     ;SKIP TO WRITE OPERATION
 1090
 1091                                 ;;*****************************************************************
 1092                                 ;LOOP #2        WRITE,WRITE CHECK
 1093
 1094 017614                          70$:
 1095
 1096
 1097                                 ;PREPARE DEVICE FOR WRITE OPERATION
 1098 017614  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      017620  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                               ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                               ;VERIFY PACK ACKNOWLEDGE
                                               ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                               ;VERIFY RECALIBRATION
      017622  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
      017624  000240                          NOP                     ;RETURN HERE IF ERROR
      017626  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      017630  000137  020250                  JMP     350$            ;GO TO 350$ IF ERROR
 1099 017634                          80$:
 1100
 1101                                 ;WRITE DATA TO THE DRIVE
 1102 017634                          120$:
 1103 017634  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 1104 017642  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 1105 017646  112722  000006                  MOVB    #RMDA,(R2)+
 1106 017652  112722  000034                  MOVB    #RMDC,(R2)+
 1107 017656  112722  000032                  MOVB    #RMOF,(R2)+
 1108 017662  112722  000004                  MOVB    #RMBA,(R2)+
 1109 017666  112722  000002                  MOVB    #RMWC,(R2)+
 1110 017672  112722  000000                  MOVB    #RMCS1,(R2)+
 1111 017676  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 1112
 1113 017702  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      017706  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
      017710  000240                          NOP                     ;RETURN HERE IF ERROR
      017712  104000                          EMT                     ;ERROR # DEFINFD BY PUT SUBROUTINE
      017714  000137  020250                  JMP     350$            ;GO TO 350$ IF ERROR
 1114 017720                          130$:
 1115
 1116                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      017720  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 1117
 1118                                 ;WAIT FOR COMMAND TO COMPLETE
      017724  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 1119
 1120                                 ;GO READ STATUS FOR WRITE COMMAND
 1121 017730  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      017734  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
      017736  000240                          NOP                     ;RETURN HERE IF ERROR
      017740  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
```

```
      017742  000137  020250                    JMP     350$            ;GO TO 350$ IF ERROR
 1122 017746                              140$:
 1123
 1124                                     ;CHECK FOR ERRORS DURING WRITE OPERATION
 1125 017746  004737  045226                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      017752  000405                            BR      150$            ;GO TO 150$ IF NO ERROR
      017754  000240                            NOP                     ;RETURN HERE IF ERROR
      017756  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      017760  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      017762  000137  020250                    JMP     350$            ;GO TO 350$ IF ERROR
 1126 017766                              150$:
 1127 017766  004737  057742                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      017772  000405                            BR      160$            ;GO TO 160$ IF NO ERROR
      017774  000240                            NOP                     ;RETURN HERE IF ERROR
      017776  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      020000  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      020002  000137  020250                    JMP     350$            ;GO TO 350$ IF ERROR
 1128 020006                              160$:
 1129 020006  004737  046060                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      020012  000405                            BR      180$            ;GO TO 180$ IF NO ERROR
      020014  000240                            NOP                     ;RETURN HERE IF ERROR
      020016  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      020020  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      020022  000137  020250                    JMP     350$            ;GO TO 350$ IF ERROR
 1130 020026                              180$:
 1131
 1132                                     ;CHANGE LOOP ADDRESSES
 1133 020026  012737  020036  001124            MOV     #190$,$LPERR
 1134 020034  000410                            BR      200$                    ;SKIP TO NEXT OPERATION
 1135
 1136                                     ;;*****************************************************************
 1137                                     ;LOOP #3        WRITE CHECK
 1138
 1139 020036                              190$:
 1140
 1141                                     ;PREPARE DEVICE FOR READ OPERATION
 1142 020036  004737  040472                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      020042  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                         ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                         ;CLEAR CONTROLLER & SELECT DEVICE
                                                                         ;VERIFY CONTROLLER CLEAR OPERATION
                                                                         ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                         ;VERIFY PACK ACKNOWLEDGE
                                                                         ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                         ;VERIFY RECALIBRATION
      020044  000404                            BR      200$            ;GO TO 200$ IF NO ERROR
      020046  000240                            NOP                     ;RETURN HERE IF ERROR
      020050  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      020052  000137  020250                    JMP     350$            ;GO TO 350$ IF ERROR
 1143 020056                              200$:
 1144
 1145                                     ;READ DATA FROM DEVICE
 1146 020056                              240$:
 1147 020056  012737  000051  001410            MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
 1148 020064  012702  001551                    MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 1149 020070  112722  000006                    MOVB    #RMDA,(R2)+
 1150 020074  112722  000032                    MOVB    #RMOF,(R2)+
```

```
                                        MOVB    #RMDC,(R2)+
                                        MOVB    #RMBA,(R2)+
                                        MOVB    #RMWC,(R2)+
                                        MOVB    #RMCS1,(R2)+
                                        MOVB    #200,(R2)

                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      250$            ;GO TO 250$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                250$:

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

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

                                ;GO READ STATUS FOR READ OPERATION
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      260$            ;GO TO 260$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                260$:

                                ;CHECK FOR ERRORS DURING READ OPERATION
                                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                                        BR      270$            ;GO TO 270$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                270$:
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      280$            ;GO TO 280$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                280$:
                                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                                        BR      300$            ;GO TO 300$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                300$:

                                350$:

                                ;;*****************************************************************
                                ;*TEST 13       WRITE, READ W/ IMPLIED SEEK
                                ;;*****************************************************************
                                TST13:
                                        SCOPE                   ;SCOPE CALL
```

```
        020252  000240                          NOP                     ;START OF TEST
        020254  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        020260  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        020264  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        020270  012737  000013  001226          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  1179
  1180                                  ;;*****************************************************************
  1181                                  ;LOOP #1        FORMAT,WRITE,READ
  1182
  1183  020276                          10$:
  1184
  1185                                  ;PREPARE THE DEVICE FOR FORMAT OPERATION
  1186  020276  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        020302  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                        ;VERIFY RECALIBRATION
        020304  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        020306  000240                          NOP                     ;RETURN HERE IF ERROR
        020310  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        020312  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
  1187
  1188                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  1189  020316                          20$:
  1190  020316  012737  000000  001444          MOV     #0,RMDCO                ;CYLINDER = 0
  1191  020324  012737  000000  001416          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
  1192  020332  013737  001444  021472          MOV     RMDCO,360$              ;SAVE DESIRED CYLINDER
  1193  020340  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
  1194  020346  012737  177376  001412          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
  1195  020354  012737  010000  001442          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
  1196  020362  012737  000063  001410          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
  1197
  1198                                  ;VERIFY THAT SECTOR IS NOT BAD
        020370  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        020374  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
        020376  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        020402  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        020404  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
  1199  020410                          25$:
  1200  020410  012737  071666  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
  1201  020416  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
  1202  020424  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
  1203
  1204                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
  1205  020430  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
  1206  020434  112722  000034                  MOVB    #RMDC,(R2)+
  1207  020440  112722  000006                  MOVB    #RMDA,(R2)+
  1208  020444  112722  000004                  MOVB    #RMBA,(R2)+
  1209  020450  112722  000002                  MOVB    #RMWC,(R2)+
  1210  020454  112722  000032                  MOVB    #RMOF,(R2)+
  1211  020460  112722  000000                  MOVB    #RMCS1,(R2)+
  1212  020464  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
  1213  020470                          30$:
```

```
                                ;FORMAT THE DRIVE
020470  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
020474  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
020476  000240                          NOP                     ;RETURN HERE IF ERROR
020500  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
020502  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
020506                          40$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
020506  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                ;WAIT FOR COMMAND TO COMPLETE
020512  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                ;GO READ STATUS FOR FORMAT OPERATION
020516  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
020522  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
020524  000240                          NOP                     ;RETURN HERE IF ERROR
020526  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
020530  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
020534                          50$:

                                ;VERIFY NO ERRORS DURING FORMAT
020534  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
020540  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
020542  000240                          NOP                     ;RETURN HERE IF ERROR
020544  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
020546  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
020550  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
020554                          60$:

                                ;MOVE LOOP ADDRESSES TO NEXT OPERATION
020554  012737  020564  001124          MOV     #70$,$LPERR
020562  000410                          BR      80$                     ;SKIP TO WRITE OPERATION

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

020564                          70$:

                                ;PPEPARE DEVICE FOR WRITE OPERATION
020564  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
020570  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
020572  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
020574  000240                          NOP                     ;RETURN HERE IF ERROR
020576  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
020600  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
020604                          80$:
020604  012737  001466  001444          MOV     #822.,RMDCO             ;SEEK TO LAST CYLINDER
```

```
1244 020612  012737  000005  001410          MOV     #SEEK.GO,RMCS10         ;WRITE SEEK COMMAND
1245
1246 020620  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     020624  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
     020626  000240                          NOP                     ;RETURN HERE IF ERROR
     020630  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     020632  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
1247 020636                          90$:
1248
1249                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     020636  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1250
1251                                 ;WAIT FOR COMMAND TO COMPLETE
     020642  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1252
1253                                 ;GO GET REGISTER STATUS
1254 020646  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     020652  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
     020654  000240                          NOP                     ;RETURN HERE IF ERROR
     020656  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     020660  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
1255 020664                          100$:
1256
1257                                 ;VERIFY RESULTS OF SEEK COMMAND
1258 020664  004737  052344                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     020670  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
     020672  000240                          NOP                     ;RETURN HERE IF ERROR
     020674  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     020676  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     020700  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
1259 020704                          110$:
1260
1261                                 ;SETUP PARAMETERS AND EXCUTE WRITE DATA COMMAND
1262 020704                          120$:
1263 020704  013737  021472  001444          MOV     360$,RMDCO              ;RESTORE CYLINDER
1264 020712  012737  177400  001412          MOV     #-256.,RMWCO            ;CHANGE WORD COUNT
1265 020720  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE BUS ADDRESS
1266 020726  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
1267 020734  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1268 020740  112722  000006                  MOVB    #RMDA,(R2)+
1269 020744  112722  000034                  MOVB    #RMDC,(R2)+
1270 020750  112722  000032                  MOVB    #RMOF,(R2)+
1271 020754  112722  000004                  MOVB    #RMBA,(R2)+
1272 020760  112722  000002                  MOVB    #RMWC,(R2)+
1273 020764  112722  000000                  MOVB    #RMCS1,(R2)+
1274 020770  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
1275
1276 020774  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021000  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
     021002  000240                          NOP                     ;RETURN HERE IF ERROR
     021004  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021006  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
1277 021012                          130$:
1278
1279                                 ;WAIT FOR COMMAND TO COMPLETE
     021012  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1280
```

```
                                ;GO READ STATUS FOR WRITE COMMAND
021016  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
021022  000404                        BR      140$            ;GO TO 140$ IF NO ERROR
021024  000240                        NOP                     ;RETURN HERE IF ERROR
021026  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
021030  000137  021470                JMP     350$            ;GO TO 350$ IF ERROR
021034                          140$:

                                ;CHECK FOR ERRORS DURING WRITE OPERATION
021034  004737  045226                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
021040  000405                        BR      150$            ;GO TO 150$ IF NO ERROR
021042  000240                        NOP                     ;RETURN HERE IF ERROR
021044  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
021046  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021050  000137  021470                JMP     350$            ;GO TO 350$ IF ERROR
021054                          150$:
021054  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
021060  000405                        BR      160$            ;GO TO 160$ IF NO ERROR
021062  000240                        NOP                     ;RETURN HERE IF ERROR
021064  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
021066  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021070  000137  021470                JMP     350$            ;GO TO 350$ IF ERROR
021074                          160$:
021074  004737  046060                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
021100  000405                        BR      180$            ;GO TO 180$ IF NO ERROR
021102  000240                        NOP                     ;RETURN HERE IF ERROR
021104  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
021106  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021110  000137  021470                JMP     350$            ;GO TO 350$ IF ERROR
021114                          180$:

                                ;CHANGE LOOP ADDRESSES
021114  012737  021124  001124        MOV     #190$,$LPERR
021122  000410                        BR      200$                    ;SKIP TO NEXT OPERATION

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

021124                          190$:

                                ;PREPARE DEVICE FOR READ OPERATION
021124  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
021130  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
021132  000404                        BR      200$            ;GO TO 200$ IF NO ERROR
021134  000240                        NOP                     ;RETURN HERE IF ERROR
021136  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
021140  000137  021470                JMP     350$            ;GO TO 350$ IF ERROR
021144                          200$:
021144  012737  001466  001444        MOV     #822.,RMDC0             ;SEEK TO LAST CYLINDER
021152  012737  000005  001410        MOV     #SEEK.GO,RMCS10         ;WRITE SEEK COMMAND
```

```
1307
1308 021160  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021164  000404                  BR      210$            ;GO TO 210$ IF NO ERROR
     021166  000240                  NOP                     ;RETURN HERE IF ERROR
     021170  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021172  000137  021470          JMP     350$            ;GO TO 350$ IF ERROR
1309 021176                      210$:
1310
1311                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     021176  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1312
1313                             ;WAIT FOR COMMAND TO COMPLETE
     021202  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1314
1315                             ;GO GET REGISTER STATUS
1316 021206  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021212  000404                  BR      220$            ;GO TO 220$ IF NO ERROR
     021214  000240                  NOP                     ;RETURN HERE IF ERROR
     021216  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021220  000137  021470          JMP     350$            ;GO TO 350$ IF ERROR
1317 021224                      220$:
1318
1319                             ;VERIFY RESUTS OF SEEK COMMAND
1320 021224  004737  052344          JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     021230  000405                  BR      230$            ;GO TO 230$ IF NO ERROR
     021232  000240                  NOP                     ;RETURN HERE IF ERROR
     021234  104000                  EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     021236  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021240  000137  021470          JMP     350$            ;GO TO 350$ IF ERROR
1321 021244                      230$:
1322
1323                             ;SETUP PARAMETERS AND EXCUTE READ DATA COMMAND
1324 021244                      240$:
1325 021244  013737  021472  001444      MOV     360$,RMDCO              ;RESTORE CYLINDER
1326 021252  012737  000071  001410      MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
1327 021260  012737  107324  001414      MOV     #BUFTWO+4,RMBA0         ;LOAD STARTING BUFFER ADDRESS
1328 021266  012702  001551              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1329 021272  112722  000006              MOVB    #RMDA,(R2)+
1330 021276  112722  000032              MOVB    #RMOF,(R2)+
1331 021302  112722  000034              MOVB    #RMDC,(R2)+
1332 021306  112722  000004              MOVB    #RMBA,(R2)+
1333 021312  112722  000002              MOVB    #RMWC,(R2)+
1334 021316  112722  000000              MOVB    #RMCS1,(R2)+
1335 021322  112712  000200              MOVB    #200,(R2)
1336
1337 021326  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021332  000404                  BR      250$            ;GO TO 250$ IF NO ERROR
     021334  000240                  NOP                     ;RETURN HERE IF ERROR
     021336  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021340  000137  021470          JMP     350$            ;GO TO 350$ IF ERROR
1338 021344                      250$:
1339
1340                             ;WAIT FOR COMMAND TO COMPLETE
     021344  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1341
1342                             ;GO READ STATUS FOR READ OPERATION
1343 021350  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
```

```
                021354  000404                          BR      260$            ;GO TO 260$ IF NO ERROR
                021356  000240                          NOP                     ;RETURN HERE IF ERROR
                021360  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                021362  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
    1344        021366                          260$:
    1345
    1346                                        ;CHECK FOR ERRORS DURING READ OPERATION
    1347        021366  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                021372  000405                          BR      270$            ;GO TO 270$ IF NO ERROR
                021374  000240                          NOP                     ;RETURN HERE IF ERROR
                021376  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                021400  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                021402  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
    1348        021406                          270$:
    1349        021406  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                021412  000405                          BR      280$            ;GO TO 280$ IF NO ERROR
                021414  000240                          NOP                     ;RETURN HERE IF ERROR
                021416  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUT'NE
                021420  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                021422  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
    1350        021426                          280$:
    1351        021426  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                021432  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
                021434  000240                          NOP                     ;RETURN HERE IF ERROR
                021436  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                021440  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                021442  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
    1352        021446                          300$:
    1353        021446  004737  043602                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
                021452  106320                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
                021454  107324                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
                021456  000404                          BR      310$            ;GO TO 310$ IF NO ERROR
                021460  000240                          NOP                     ;RETURN HERE IF ERROR
                021462  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
                021464  000137  021470                  JMP     350$            ;GO TO 350$ IF ERROR
    1354        021470                          310$:
    1355
    1356        021470                          350$:
    1357        021470  000401                          BR      370$
    1358
    1359        021472  000000                  360$:   .WORD   0
    1360
    1361        021474                          370$:
    1362
    1363                                        ;;*****************************************************************
                                                ;*TEST 14        WRITE, WRITE CHECK W/ HEAD SWITCHING
                                                ;;*****************************************************************
                021474                          TST14:
                021474  000004                          SCOPE                   ;SCOPE CALL
                021476  000240                          NOP                     ;START OF TEST
                021500  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                021504  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                021510  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                021514  012737  000014  001226          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    1364
    1365                                        ;;*****************************************************************
    1366                                        ;LOOP #1        FORMAT,WRITE,WRITE CHECK
```

```
1367
1368                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1369 021522                                10$:
1370 021522  012737  000000  001444              MOV     #0,RMDCO                ;CYLINDER = 0
1371 021530  112737  000000  001417              MOVB    #0,RMDAO+1              ;TRACK = 0
1372 021536  112737  000037  001416  15$:        MOVB    #31.,RMDAO              ;SECTOR = 31.
1373 021544  012737  010000  001442              MOV     #FMT16,RMOFO            ;16 BIT FORMAT
1374 021552  012737  106314  001414              MOV     #BUFONE,RMBAO           ;BUS ADDRESS
1375 021560  012737  176774  001412              MOV     #-258.*2,RMWCO          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
1376 021566  012737  000063  001410              MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
1377
1378                                        ;VERIFY THAT SECTOR IS NOT BAD
     021574  004737  041416                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     021600  000405                                BR      20$             ;GO TO 20$ IF NO ERROR
     021602  104401  070230                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     021606  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     021610  000137  022702                        JMP     350$            ;GO TO 350$ IF ERROR
1379 021614                                20$:
1380 021614  123727  001416  000037              CMPB    RMDAO,#31.              ;IS LAST TRACK ASSIGNED ?
1381 021622  001345                            BNE     15$                     ;BR IF NO
1382 021624  012737  071666  001174            MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
1383 021632  012737  000001  001176            MOV     #1,$TMP1                ;RANGE OF PATTERN
1384 021640  004737  043344                    JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
1385
1386                                        ;PREPARE THE DEVICE FOR FORMAT OPERATION
1387 021644  004737  040472                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     021650  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
     021652  000404                            BR      30$             ;GO TO 30$ IF NO ERROR
     021654  000240                            NOP                     ;RETURN HERE IF ERROR
     021656  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     021660  000137  022702                    JMP     350$            ;GO TO 350$ IF ERROR
1388 021664                                30$:
1389
1390                                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
1391 021664  012702  001551                    MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
1392 021670  112722  000034                    MOVB    #RMDC,(R2)+
1393 021674  112722  000006                    MOVB    #RMDA,(R2)+
1394 021700  112722  000004                    MOVB    #RMBA,(R2)+
1395 021704  112722  000002                    MOVB    #RMWC,(R2)+
1396 021710  112722  000032                    MOVB    #RMOF,(R2)+
1397 021714  112722  000000                    MOVB    #RMCS1,(R2)+
1398 021720  112712  000200                    MOVB    #200,(R2)               ;TERMINATE TABLE
1399
1400                                        ;FORMAT THE DRIVE
1401 021724  004737  044500                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021730  000404                            BR      40$             ;GO TO 40$ IF NO ERROR
     021732  000240                            NOP                     ;RETURN HERE IF ERROR
     021734  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021736  000137  022702                    JMP     350$            ;GO TO 350$ IF ERROR
1402 021742                                40$:
```

```
1403
1404                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
021742  004737  044144                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1405
1406                                    ;WAIT FOR COMMAND TO COMPLETE
021746  004737  045042                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1407
1408                                    ;GO READ STATUS FOR FORMAT OPERATION
1409 021752  004737  044230                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021756  000404                            BR      50$             ;GO TO 50$ IF NO ERROR
     021760  000240                            NOP                     ;RETURN HERE IF ERROR
     021762  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021764  000137  022702                    JMP     350$            ;GO TO 350$ IF ERROR
1410 021770                             50$:
1411
1412                                    ;VERIFY NO ERRORS DURING FORMAT
1413 021770  004737  057742                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     021774  000405                            BR      60$             ;GO TO 60$ IF NO ERROR
     021776  000240                            NOP                     ;RETURN HERE IF ERROR
     022000  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     022002  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022004  000137  022702                    JMP     350$            ;GO TO 350$ IF ERROR
1414 022010                             60$:
1415
1416                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1417 022010  012737  022062  001124            MOV     #70$,$LPERR
1418
1419                                    ;REGENERATE BUFFER
1420 022016  012737  177000  001412            MOV     #-256.*2,RMWCO          ;CHANGE WORD COUNT
1421 022024  012737  106314  001414            MOV     #BUFONE,RMBAO           ;CHANGE BUS ADDRESS
1422 022032  012737  071730  001174            MOV     #ZEROS,$TMP0            ;STARTING ADDRESS
1423 022040  012737  000001  001176            MOV     #1,$TMP1                ;RANGE
1424 022046  012737  000060  001410            MOV     #WD,RMCS1O              ;WRITE DATA
1425 022054  004737  043344                    JSR     PC,GENBUF               ;GENERATE BUFFER
1426 022060  000410                            BR      80$                     ;SKIP TO WRITE OPERATION
1427
1428                                    ;;*****************************************************************
1429                                    ;LOOP #2        WRITE,WRITE CHECK
1430
1431 022062                             70$:
1432
1433                                    ;PREPARE DEVICE FOR WRITE OPERATION
1434 022062  004737  040472                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     022066  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR ''PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
     022070  000404                            BR      80$             ;GO TO 80$ IF NO ERROR
     022072  000240                            NOP                     ;RETURN HERE IF ERROR
     022074  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     022076  000137  022702                    JMP     350$            ;GO TO 350$ IF ERROR
1435 022102                             80$:
1436 022102  012737  000005  001410            MOV     #SEEK!GO,RMCS1O         ;LOAD SEEK COMMAND
```

```
1437
1438 022110  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022114  000404                  BR      90$             ;GO TO 90$ IF NO ERROR
     022116  000240                  NOP                     ;RETURN HERE IF ERROR
     022120  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     022122  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1439 022126                      90$:
1440
1441                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     022126  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1442
1443                             ;WAIT FOR COMMAND TO COMPLETE
     022132  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1444
1445                             ;GO GET REGISTER STATUS
1446 022136  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022142  000404                  BR      100$            ;GO TO 100$ IF NO ERROR
     022144  000240                  NOP                     ;RETURN HERE IF ERROR
     022146  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     022150  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1447 022154                      100$:
1448
1449                             ;VERIFY RESULTS OF SEEK COMMAND
1450 022154  004737  052344          JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     022160  000405                  BR      110$            ;GO TO 110$ IF NO ERROR
     022162  000240                  NOP                     ;RETURN HERE IF ERROR
     022164  104000                  EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     022166  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022170  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1451 022174                      110$:
1452
1453                             ;WRITE DATA TO THE DRIVE
1454 022174                      120$:
1455 022174  012737  000061  001410      MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
1456 022202  012702  001551              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1457 022206  112722  000006              MOVB    #RMDA,(R2)+
1458 022212  112722  000034              MOVB    #RMDC,(R2)+
1459 022216  112722  000032              MOVB    #RMOF,(R2)+
1460 022222  112722  000004              MOVB    #RMBA,(R2)+
1461 022226  112722  000002              MOVB    #RMWC,(R2)+
1462 022232  112722  000000              MOVB    #RMCS1,(R2)+
1463 022236  112722  000200              MOVB    #200,(R2)+              ;TERMINATE TABLE
1464
1465 022242  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     022246  000404                  BR      130$            ;GO TO 130$ IF NO ERROR
     022250  000240                  NOP                     ;RETURN HERE IF ERROR
     022252  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     022254  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1466 022260                      130$:
1467
1468                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     022260  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1469
1470                             ;WAIT FOR COMMAND TO COMPLETE
     022264  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1471
1472                             ;GO READ STATUS FOR WRITE COMMAND
```

```
1473 022270  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     022274  000404                  BR      140$            ;GO TO 140$ IF NO ERROR
     022276  000240                  NOP                     ;RETURN HERE IF ERROR
     022300  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     022302  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1474 022306                  140$:
1475
1476                         ;CHECK FOR ERRORS DURING WRITE OPERATION
1477 022306  004737  045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     022312  000405                  BR      150$            ;GO TO 150$ IF NO ERROR
     022314  000240                  NOP                     ;RETURN HERE IF ERROR
     022316  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     022320  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022322  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1478 022326                  150$:
1479 022326  004737  057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     022332  000405                  BR      160$            ;GO TO 160$ IF NO ERROR
     022334  000240                  NOP                     ;RETURN HERE IF ERROR
     022336  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     022340  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022342  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1480 022346                  160$:
1481 022346  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     022352  000405                  BR      180$            ;GO TO 180$ IF NO ERROR
     022354  000240                  NOP                     ;RETURN HERE IF ERROR
     022356  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     022360  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     022362  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1482 022366                  180$:
1483
1484                         ;CHANGE LOOP ADDRESSES
1485 022366  012737  022376  001124          MOV     #190$,$LPERR
1486 022374  000410                  BR      200$                    ;SKIP TO NEXT OPERATION
1487
1488                         ;;*****************************************************************
1489                         ;LOOP #3        WRITE CHECK
1490
1491 022376                  190$:
1492
1493                         ;PREPARE DEVICE FOR READ OPERATION
1494 022376  004737  040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     022402  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                            ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                            ;VERIFY PACK ACKNOWLEDGE
                                                            ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                            ;VERIFY RECALIBRATION
     022404  000404                  BR      200$            ;GO TO 200$ IF NO ERROR
     022406  000240                  NOP                     ;RETURN HERE IF ERROR
     022410  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     022412  000137  022702          JMP     350$            ;GO TO 350$ IF ERROR
1495 022416                  200$:
1496 022416  012737  000005  001410          MOV     #SEEK!GO,RMCS10         ;LOAD SEEK COMMAND
1497
1498 022424  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
```

```
        022430  000404                          BR      210$            ;GO TO 210$ IF NO ERROR
        022432  000240                          NOP                     ;RETURN HERE IF ERROR
        022434  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        022436  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
  1499  022442                          210$:
  1500
  1501                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        022442  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  1502
  1503                                  ;WAIT FOR COMMAND TO COMPLETE
        022446  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  1504
  1505                                  ;GO READ REGISTER STATUS
  1506  022452  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        022456  000404                          BR      220$            ;GO TO 220$ IF NO ERROR
        022460  000240                          NOP                     ;RETURN HERE IF ERROR
        022462  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        022464  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
  1507  022470                          220$:
  1508
  1509                                  ;GO VERIFY RESULTS OF SEEK
  1510  022470  004737  052344                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        022474  000405                          BR      230$            ;GO TO 230$ IF NO ERROR
        022476  000240                          NOP                     ;RETURN HERE IF ERROR
        022500  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        022502  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022504  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
  1511  022510                          230$:
  1512
  1513                                  ;READ DATA FROM DEVICE
  1514  022510                          240$:
  1515  022510  012737  000051  001410          MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
  1516  022516  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  1517  022522  112722  000006                  MOVB    #RMDA,(R2)+
  1518  022526  112722  000032                  MOVB    #RMOF,(R2)+
  1519  022532  112722  000034                  MOVB    #RMDC,(R2)+
  1520  022536  112722  000004                  MOVB    #RMBA,(R2)+
  1521  022542  112722  000002                  MOVB    #RMWC,(R2)+
  1522  022546  112722  000000                  MOVB    #RMCS1,(R2)+
  1523  022552  112712  000200                  MOVB    #200,(R2)
  1524
  1525  022556  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        022562  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
        022564  000240                          NOP                     ;RETURN HERE IF ERROR
        022566  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        022570  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
  1526  022574                          250$:
  1527
  1528                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        022574  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  1529
  1530                                  ;WAIT FOR COMMAND TO COMPLETE
        022600  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  1531
  1532                                  ;GO READ STATUS FOR READ OPERATION
  1533  022604  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        022610  000404                          BR      260$            ;GO TO 260$ IF NO ERROR
```

```
                022612  000240                          NOP                     ;RETURN HERE IF ERROR
                022614  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                022616  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
   1534         022622                          260$:
   1535
   1536                                         ;CHECK FOR ERRORS DURING READ OPERATION
   1537         022622  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                022626  000405                          BR      270$            ;GO TO 270$ IF NO ERROR
                022630  000240                          NOP                     ;RETURN HERE IF ERROR
                022632  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                022634  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022636  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
   1538         022642                          270$:
   1539         022642  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                022646  000405                          BR      280$            ;GO TO 280$ IF NO ERROR
                022650  000240                          NOP                     ;RETURN HERE IF ERROR
                022652  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                022654  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022656  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
   1540         022662                          280$:
   1541         022662  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                022666  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
                022670  000240                          NOP                     ;RETURN HERE IF ERROR
                022672  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                022674  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022676  000137  022702                  JMP     350$            ;GO TO 350$ IF ERROR
   1542         022702                          300$:
   1543
   1544         022702                          350$:
   1545
   1546                                         ;;*****************************************************************
                                                ;*TEST 15        WRITE, WRITE CHECK W/ MID-TRANSFER SEEK
                                                ;;*****************************************************************
                022702                          TST15:
                022702  000004                          SCOPE                   ;SCOPE CALL
                022704  000240                          NOP                     ;START OF TEST
                022706  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                022712  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                022716  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                022722  012737  000015  001226          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   1547
   1548                                         ;;*****************************************************************
   1549                                         ;LOOP #1        FORMAT,WRITE,WRITE CHECK
   1550
   1551                                         ;LOAD PARAMETERS AND GENERATE DATA BUFFER
   1552         022730                          10$:
   1553         022730  012737  000000  001444          MOV     #0,RMDC0                ;CYLINDER = 0
   1554         022736  013737  001332  001416  15$:    MOV     LSTRK,RMDA0             ;SET LAST TRACK AND
   1555         022744  112737  000037  001416          MOVB    #31.,RMDA0              ;LAST SECTOR
   1556         022752  012737  010000  001442          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
   1557         022760  012737  106314  001414          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
   1558         022766  012737  176774  001412          MOV     #-258.*2,RMWC0          ;WORD COUNT FOR 2 SECTORS (2' COMP)
   1559         022774  012737  000063  001410          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
   1560
   1561                                         ;VERIFY THAT SECTOR IS NOT BAD
                023002  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
                023006  000405                          BR      20$             ;GO TO 20$ IF NO ERROR
```

```
        023010  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        023014  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        023016  000137  023724                  JMP     350$            ;GO TO 350$ IF ERROR
   1562 023022                          20$:
   1563 023022  123737  001417  001333          CMPB    RMDA0+1,LSTRK+1         ;IS LAST TRACK ASSIGNED ?
   1564 023030  001342                          BNE     15$             ;BR IF NO
   1565 023032  012737  071730  001174          MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
   1566 023040  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
   1567 023046  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
   1568
   1569                                 ;PREPARE THE DEVICE FOR FORMAT OPERATION
   1570 023052  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        023056  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
        023060  000404                          BR      25$             ;GO TO 25$ IF NO ERROR
        023062  000240                          NOP                     ;PETURN HERE IF ERROR
        023064  104000                          EMT                     ;ERROR # DEFINED BY ISTPRP SUBROUTINE
        023066  000137  023724                  JMP     350$            ;GO TO 350$ IF ERROR
   1571 023072                          25$:
   1572
   1573                                 ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
   1574 023072  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
   1575 023076  112722  000034                  MOVB    #RMDC,(R2)+
   1576 023102  112722  000006                  MOVB    #RMDA,(R2)+
   1577 023106  112722  000004                  MOVB    #RMBA,(R2)+
   1578 023112  112722  000002                  MOVB    #RMWC,(R2)+
   1579 023116  112722  000032                  MOVB    #RMOF,(R2)+
   1580 023122  112722  000000                  MOVB    #RMCS1,(R2)+
   1581 023126  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
   1582 023132                          30$:
   1583 023132  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        023136  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        023140  000240                          NOP                     ;RETURN HERE IF ERROR
        023142  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        023144  000137  023724                  JMP     350$            ;GO TO 350$ IF ERROR
   1584 023150                          40$:
   1585
   1586                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        023150  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
   1587
   1588                                 ;WAIT FOR COMMAND TO COMPLETE
        023154  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
   1589
   1590                                 ;GO READ STATUS FOR FORMAT OPERATION
   1591 023160  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        023164  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        023166  000240                          NOP                     ;RETURN HERE IF ERROR
        023170  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        023172  000137  023724                  JMP     350$            ;GO TO 350$ IF ERROR
   1592 023176                          50$:
   1593
```

```
                                        ;VERIFY NO ERRORS DURING FORMAT
023176  004737  057742                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
023202  000405                              BR      60$             ;GO TO 60$ IF NO ERROR
023204  000240                              NOP                     ;RETURN HERE IF ERROR
023206  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
023210  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
023212  000137  023724                      JMP     350$            ;GO TO 350$ IF ERROR
023216                                  60$:

                                        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
023216  012737  023270  001124                MOV     #70$,$LPERR

                                        ;REGENERATE BUFFER
023224  012737  000060  001410                MOV     #WD,RMCS10              ;WRITE DATA
023232  012737  177000  001412                MOV     #-256.*2,RMWCO  ;CHANGE WORD COUNT
023240  012737  106314  001414                MOV     #BUFONE,RMBAO           ;CHANGE BUS ADDRSS
023246  012737  071666  001174                MOV     #ONES,$TMPO             ;PATTERN ADDRESS
023254  012737  000001  001176                MOV     #1,$TMP1                ;PATTERN RANGE
023262  004737  043344                        JSR     PC,GENBUF
023266  000410                              BR      80$                     ;SKIP TO WRITE OPERATION

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

                                        70$:

                                        ;PREPARE DEVICE FOR WRITE OPERATION
023270  004737  040472                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
023274  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
023276  000404                              BR      80$             ;GO TO 80$ IF NO ERROR
023300  000240                              NOP                     ;RETURN HERE IF ERROR
023302  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
023304  000137  023724                      JMP     350$            ;GO TO 350$ IF ERROR
023310                                  80$:

                                        ;WRITE DATA TO THE DRIVE
023310                                  120$:
023310  012737  000061  001410                MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
023316  012702  001551                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
023322  112722  000006                        MOVB    #RMDA,(R2)+
023326  112722  000034                        MOVB    #RMDC,(R2)+
023332  112722  000032                        MOVB    #RMOF,(R2)+
023336  112722  000004                        MOVB    #RMBA,(R2)+
023342  112722  000002                        MOVB    #RMWC,(R2)+
023346  112722  000000                        MOVB    #RMCS1,(R2)+
023352  112722  000200                        MOVB    #200,(R2)+              ;TERMINATE TABLE

023356  004737  044500                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
023362  000404                              BR      130$            ;GO TO 130$ IF NO ERROR
023364  000240                              NOP                     ;RETURN HERE IF ERROR
```

```
        023366  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        023370  000137  023724          JMP     350$            ;GO TO 350$ IF ERROR
  1632  023374                  130$:
  1633
  1634                          ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        023374  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  1635
  1636                          ;WAIT FOR COMMAND TO COMPLETE
        023400  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  1637
  1638                          ;GO READ STATUS FOR WRITE COMMAND
  1639  023404  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        023410  000404                  BR      140$            ;GO TO 140$ IF NO ERROR
        023412  000240                  NOP                     ;RETURN HERE IF ERROR
        023414  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        023416  000137  023724          JMP     350$            ;GO TO 350$ IF ERROR
  1640  023422                  140$:
  1641
  1642                          ;CHECK FOR ERRORS DURING WRITE OPERATION
  1643  023422  004737  045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        023426  000405                  BR      150$            ;GO TO 150$ IF NO ERROR
        023430  000240                  NOP                     ;RETURN HERE IF ERROR
        023432  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        023434  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023436  000137  023724          JMP     350$            ;GO TO 350$ IF ERROR
  1644  023442                  150$:
  1645  023442  004737  057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        023446  000405                  BR      160$            ;GO TO 160$ IF NO ERROR
        023450  000240                  NOP                     ;RETURN HERE IF ERROR
        023452  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        023454  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023456  000137  023724          JMP     350$            ;GO TO 350$ IF ERROR
  1646  023462                  160$:
  1647  023462  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        023466  000405                  BR      180$            ;GO TO 180$ IF NO ERROR
        023470  000240                  NOP                     ;RETURN HERE IF ERROR
        023472  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        023474  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023476  000137  023724          JMP     350$            ;GO TO 350$ IF ERROR
  1648  023502                  180$:
  1649
  1650                          ;CHANGE LOOP ADDRESSES
  1651  023502  012737  023512  001124  MOV     #190$,$LPERR
  1652  023510  000410                  BR      200$                    ;SKIP TO NEXT OPERATION
  1653
  1654                          ;;*****************************************************************
  1655                          ;LOOP #3        WRITE CHECK
  1656
  1657  023512                  190$:
  1658
  1659                          ;PREPARE DEVICE FOR READ OPERATION
  1660  023512  004737  040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        023516  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
023520  000404                  BR      200$          ;GO TO 200$ IF NO ERROR
023522  000240                  NOP                   ;RETURN HERE IF ERROR
023524  104000                  EMT                   ;ERROR # DEFINED BY TSTPRP SUBROUTINE
023526  000137  023724          JMP     350$          ;GO TO 350$ IF ERROR
1661 023532             200$:
1662
1663                    ;READ DATA FROM DEVICE
1664 023532             240$:
1665 023532  012737  000051  001410  MOV     #WCD!GO,RMCS10        ;READ DATA COMMAND
1666 023540  012702  001551          MOV     #PUTINX,R2            ;LOAD PUT REGISTER INDEX TABLE
1667 023544  112722  000006          MOVB    #RMDA,(R2)+
1668 023550  112722  000032          MOVB    #RMOF,(R2)+
1669 023554  112722  000034          MOVB    #RMDC,(R2)+
1670 023560  112722  000004          MOVB    #RMBA,(R2)+
1671 023564  112722  000002          MOVB    #RMWC,(R2)+
1672 023570  112722  000000          MOVB    #RMCS1,(R2)+
1673 023574  112712  000200          MOVB    #200,(R2)
1674
1675 023600  004737  044500          JSR     PC,PUT        ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
023604  000404                  BR      250$          ;GO TO 250$ IF NO ERROR
023606  000240                  NOP                   ;RETURN HERE IF ERROR
023610  104000                  EMT                   ;ERROR # DEFINED BY PUT SUBROUTINE
023612  000137  023724          JMP     350$          ;GO TO 350$ IF ERROR
1676 023616             250$:
1677
1678                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
023616  004737  044144          JSR     PC,GETSTS     ;GO TO GETSTS SUBROUTINE
1679
1680                    ;WAIT FOR COMMAND TO COMPLETE
023622  004737  045042          JSR     PC,TIMOUT     ;GO TO TIMOUT SUBROUTINE
1681
1682                    ;GO READ STATUS FOR READ OPERATION
1683 023626  004737  044230          JSR     PC,GET        ;GO READ REGISTER(S) WITH GET SUBROUTINE
023632  000404                  BR      260$          ;GO TO 260$ IF NO ERROR
023634  000240                  NOP                   ;RETURN HERE IF ERROR
023636  104000                  EMT                   ;ERROR # DEFINED BY GET SUBROUTINE
023640  000137  023724          JMP     350$          ;GO TO 350$ IF ERROR
1684 023644             260$:
1685
1686                    ;CHECK FOR ERRORS DURING READ OPERATION
1687 023644  004737  045226          JSR     PC,PRIERR     ;GO CHECK FOR PRIMARY ERRORS
023650  000405                  BR      270$          ;GO TO 270$ IF NO ERROR
023652  000240                  NOP                   ;RETURN HERE IF ERROR
023654  104000                  EMT                   ;ERROR # DEFINED BY PRIERR SUBROUTINE
023656  004736                  JSR     PC,@(SP)+     ;GO BACK FOR MORE ERROR CHECKS
023660  000137  023724          JMP     350$          ;GO TO 350$ IF ERROR
1688 023664             270$:
1689 023664  004737  057742          JSR     PC,DTASTS     ;GO VERIFY RESULTS OF DATA TRANSFER
023670  000405                  BR      280$          ;GO TO 280$ IF NO ERROR
023672  000240                  NOP                   ;RETURN HERE IF ERROR
023674  104000                  EMT                   ;ERROR # DEFINED BY DTASTS SUBROUTINE
023676  004736                  JSR     PC,@(SP)+     ;GO BACK FOR MORE ERROR CHECKS
023700  000137  023724          JMP     350$          ;GO TO 350$ IF ERROR
1690 023704             280$:
```

```
1691 023704 004737 046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     023710 000405                 BR      300$            ;GO TO 300$ IF NO ERROR
     023712 000240                 NOP                     ;RETURN HERE IF ERROR
     023714 104000                 EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     023716 004736                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023720 000137 023724          JMP     350$            ;GO TO 350$ IF ERROR
1692 023724                    300$:
1693
1694 023724                    350$:
1695
1696                           ;;*****************************************************************
                               ;*TEST 16       WRITE, READ W/ HCE ERROR
                               ;;*****************************************************************
     023724                    TST16:
     023724 000004                 SCOPE                   ;SCOPE CALL
     023726 000240                 NOP                     ;START OF TEST
     023730 012706 001100          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     023734 013700 001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     023740 013701 001464          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     023744 012737 000016 001226   MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1697
1698                           ;;*****************************************************************
1699                           ;LOOP #1       FORMAT,WRITE,READ
1700
1701 023752 012704 000000          MOV     #0,R4           ;R4 = 1ST HEADER WORD
1702 023756 012703 000001      10$:MOV     #1,R3           ;R3 = HCE BIT
1703 023762                    15$:
1704
1705                           ;PREPARE THE DEVICE FOR FORMAT OPERATION
1706 023762 004737 040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     023766 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
     023770 000404                 BR      20$             ;GO TO 20$ IF NO ERROR
     023772 000240                 NOP                     ;RETURN HERE IF ERROR
     023774 104000                 EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     023776 000137 025572          JMP     370$            ;GO TO 370$ IF ERROR
1707
1708                           ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1709 024002                    20$:
1710 024002 012737 000000 001444   MOV     #0,RMDCO        ;CYLINDER = 0
1711 024010 012737 000000 001416   MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
1712 024016 012737 106314 001414   MOV     #BUFONE,RMBAO   ;BUS ADDRESS
1713 024024 012737 177376 001412   MOV     #-258.,RMWCO    ;2 + 256. WORDS (2'S COMP)
1714 024032 012737 010000 001442   MOV     #FMT16,RMOFO    ;16 BIT FORMAT
1715 024040 012737 000063 001410   MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA COMMAND
1716
1717                           ;VERIFY THAT SECTOR IS NOT BAD
     024046 004737 041416          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     024052 000405                 BR      25$             ;GO TO 25$ IF NO ERROR
     024054 104401 070230          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     024060 104000                 EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
```

```
        024062  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
   1718 024066                          25$:
   1719 024066  012737  071730  001174          MOV     #ZEROS,$TMP0    ;STARTING ADDRESS OF PATTERN
   1720 024074  012737  000001  001176          MOV     #1,$TMP1        ;RANGE OF PATTERN
   1721 024102  004737  043344                  JSR     PC,GENBUF       ;GO GENERATE BUFFER FOR FORMAT
   1722
   1723                                 ;CHANGE HEADER WORD TO FORCE HCE DURING WRITE & READ
   1724 024106  030364  106314                  BIT     R3,BUFONE(R4)   ;SET OR RESET FOR HCE??
   1725 024112  001403                          BEQ     26$
   1726 024114  040364  106314                  BIC     R3,BUFONE(R4)   ;RESET BIT FOR HCE
   1727 024120  000402                          BR      27$
   1728 024122  050364  106314          26$:    BIS     R3,BUFONE(R4)   ;SET FOR HCE
   1729
   1730                                 ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
   1731 024126                          27$:
   1732 024126  012702  001551                  MOV     #PUTINX,R2      ;R2 = BYTE ENTRY POSITION
   1733 024132  112722  000034                  MOVB    #RMDC,(R2)+
   1734 024136  112722  000006                  MOVB    #RMDA,(R2)+
   1735 024142  112722  000004                  MOVB    #RMBA,(R2)+
   1736 024146  112722  000002                  MOVB    #RMWC,(R2)+
   1737 024152  112722  000032                  MOVB    #RMOF,(R2)+
   1738 024156  112722  000000                  MOVB    #RMCS1,(R2)+
   1739 024162  112712  000200                  MOVB    #200,(R2)       ;TERMINATE TABLE
   1740 024166                          30$:
   1741 024166  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        024172  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        024174  000240                          NOP                     ;RETURN HERE IF ERROR
        024176  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        024200  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
   1742 024204                          40$:
   1743
   1744                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        024204  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
   1745
   1746                                 ;WAIT FOR COMMAND TO COMPLETE
        024210  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
   1747
   1748                                 ;GO READ STATUS FOR FORMAT OPERATION
   1749 024214  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        024220  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        024222  000240                          NOP                     ;RETURN HERE IF ERROR
        024224  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        024226  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
   1750 024232                          50$:
   1751
   1752                                 ;VERIFY NO ERRORS DURING FORMAT
   1753 024232  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        024236  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
        024240  000240                          NOP                     ;RETURN HERE IF ERROR
        024242  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        024244  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024246  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
   1754 024252                          60$:
   1755
   1756                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
   1757 024252  012737  024262  001124          MOV     #70$,$LPERR
   1758 024260  000410                          BR      80$             ;SKIP TO WRITE OPERATION
```

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

024262                                  70$:

                                        ;PREPARE DEVICE FOR WRITE OPERATION
024262  004737  040472                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
024266  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
024270  000404                                  BR      80$             ;GO TO 80$ IF NO ERROR
024272  000240                                  NOP                     ;RETURN HERE IF ERROR
024274  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
024276  000137  025420                          JMP     350$            ;GO TO 350$ IF ERROR
024302                                  80$:

                                        ;WRITE DATA TO THE DRIVE
024302                                  120$:
024302  012737  106320  001414                  MOV     #BUFONE+4,RMBAO ;CHANGE BUS ADDRESS
024310  012737  177400  001412                  MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
024316  012737  000061  001410                  MOV     #WD!GO,RMCS1O   ;WRITE DATA COMMAND
024324  012702  001551                          MOV     #PUTINX,R2      ;LOAD PUT REGISTER INDEX TABLE
024330  112722  000006                          MOVB    #RMDA,(R2)+
024334  112722  000034                          MOVB    #RMDC,(R2)+
024340  112722  000032                          MOVB    #RMOF,(R2)+
024344  112722  000004                          MOVB    #RMBA,(R2)+
024350  112722  000002                          MOVB    #RMWC,(R2)+
024354  112722  000000                          MOVB    #RMCS1,(R2)+
024360  112722  000200                          MOVB    #200,(R2)+      ;TERMINATE TABLE

024364  004737  044500                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
024370  000404                                  BR      130$            ;GO TO 130$ IF NO ERROR
024372  000240                                  NOP                     ;RETURN HERE IF ERROR
024374  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
024376  000137  025420                          JMP     350$            ;GO TO 350$ IF ERROR
024402                                  130$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
024402  004737  044144                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
024406  004737  045042                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE COMMAND
024412  004737  044230                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
024416  000404                                  BR      140$            ;GO TO 140$ IF NO ERROR
024420  000240                                  NOP                     ;RETURN HERE IF ERROR
024422  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
024424  000137  025420                          JMP     350$            ;GO TO 350$ IF ERROR
024430                                  140$:
```

```
1794                                    ;CHECK FOR ERRORS DURING WRITE OPERATION
1795 024430  004737  045226                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     024434  000405                              BR      145$            ;GO TO 145$ IF NO ERROR
     024436  000240                              NOP                     ;RETURN HERE IF ERROR
     024440  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     024442  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024444  000137  025420                      JMP     350$            ;GO TO 350$ IF ERROR
1796
1797                                    ;DECODE THE TYPE OF ERROR PRODUCED BY THE BIT POSITION OF R3
1798 024450                             145$:
1799 024450  005704                              TST     R4              ;IS THIS THE FIRST HEADER WORD ?
1800 024452  001006                              BNE     146$            ;NO !!
1801
1802 024454  032703  010000                      BIT     #FMT16,R3       ;SHOULD FER BE SET ?
1803 024460  001037                              BNE     155$            ;YES !!
1804 024462  032703  140000                      BIT     #MSE!USE,R3     ;SHOULD BSE BE SET ?
1805 024466  001061                              BNE     165$            ;YES !!
1806
1807                                    ;VERIFY THAT A HEADER COMPARE ERROR WAS DETECTED
1808 024470                             146$:
1809 024470  032737  000200  001350              BIT     #HCE,RMER1I     ;WAS HCE DETECTED ??
1810 024476  001102                              BNE     175$            ;YES !!
1811
1812 024500  004737  057742                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024504  000405                              BR      150$            ;GO TO 150$ IF NO ERROR
     024506  000240                              NOP                     ;RETURN HERE IF ERROR
     024510  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024512  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024514  000137  025420                      JMP     350$            ;GO TO 350$ IF ERROR
1813 024520                             150$:
1814 024520  013737  001350  001142              MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1815 024526  012737  000200  001140              MOV     #HCE,$GDDAT     ;EXPECTED STATUS
1816 024534  010437  001174                      MOV     R4,$TMP0        ;R4 = HEADER WORD NUMBER AND
1817 024540  001002                              BNE     151$            ;ADJUST NUMBER.
1818 024542  005237  001174                      INC     $TMP0
1819 024546  010337  001176              151$:   MOV     R3,$TMP1        ;R3 = BIT POSIITON
1820 024552  104344                              EMT     344
1821 024554  000137  025420                      JMP     350$
1822
1823                                    ;VERIFY THAT A FORMAT ERROR WAS DETECTED
1824 024560                             155$:
1825 024560  032737  000020  001350              BIT     #FER,RMER1I     ,WAS FER DETECTED ??
1826 024566  001046                              BNE     175$            ;YES !!
1827
1828 024570  004737  057742                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024574  000405                              BR      160$            ;GO TO 160$ IF NO ERROR
     024576  000240                              NOP                     ;RETURN HERE IF ERROR
     024600  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024602  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024604  000137  025420                      JMP     350$            ;GO TO 350$ IF ERROR
1829 024610                             160$:
1830 024610  012737  000020  001140              MOV     #FER,$GDDAT     ;EXPECTED STATUS
1831 024616  013737  001350  001142              MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1832 024624  104343                              EMT     343
1833 024626  000137  025420                      JMP     350$
1834
1835                                    ;VERIFY THAT A BAD SECTOR ERROR WAS DETECTED
```

```
1836 024632                                  165$:
1837 024632  032737  100000  001376                BIT     #BSE,RMER2I     ;WAS BSE DETECTED ??
1838 024640  001021                                BNE     175$            ;YES !!
1839
1840 024642  004737  057742                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024646  000405                                BR      170$            ;GO TO 170$ IF NO ERROR
     024650  000240                                NOP                     ;RETURN HERE IF ERROR
     024652  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024654  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024656  000137  025420                        JMP     350$            ;GO TO 350$ IF ERROR
1841 024662                                  170$:
1842 024662  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
1843 024670  012737  100000  001140                MOV     #BSE,$GDDAT     ;EXPECTED STAIUS
1844 024676  104345                                EMT     345
1845 024700  000137  025420                        JMP     350$
1846
1847                                         ;CHECK FOR OTHER ERRORS
1848 024704                                  175$:
1849 024704  004737  046060                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     024710  000405                                BR      180$            ;GO TO 180$ IF NO ERROR
     024712  000240                                NOP                     ;RETURN HERE IF ERROR
     024714  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     024716  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024720  000137  025420                        JMP     350$            ;GO TO 350$ IF ERROR
1850 024724                                  180$:
1851
1852                                         ;CHANGE LOOP ADDRESSES
1853 024724  012737  024732  001124                MOV     #190$,$LPERR
1854
1855                                         ;;*****************************************************************
1856                                         ;LOOP #3        READ
1857
1858 024732                                  190$:
1859
1860                                         ;PREPARE DEVICE FOR READ OPERATION
1861 024732  004737  040472                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     024736  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
     024740  000404                                BR      200$            ;GO TO 200$ IF NO ERROR
     024742  000240                                NOP                     ;RETURN HERE IF ERROR
     024744  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     024746  000137  025420                        JMP     350$            ;GO TO 350$ IF ERROR
1862 024752                                  200$:
1863
1864                                         ;READ DATA FROM DEVICE
1865 024752                                  240$:
1866 024752  012737  107324  001414                MOV     #BUFTWO+4,RMBAO ;CHANGE BUS ADDRESS
1867 024760  012737  177400  001412                MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
1868 024766  012737  000071  001410                MOV     #RD!GO,RMCS10   ;READ DATA COMMAND
1869 024774  012702  001551                        MOV     #PUTINX,R2      ;LOAD PUT REGISTER INDEX TABLE
1870 025000  112722  000006                        MOVB    #RMDA,(R2)+
```

```
 1871 025004  112722  000032                MOVB    #RMOF,(R2)+
 1872 025010  112722  000034                MOVB    #RMDC,(R2)+
 1873 025014  112722  000004                MOVB    #RMBA,(R2)+
 1874 025020  112722  000002                MOVB    #RMWC,(R2)+
 1875 025024  112722  000000                MOVB    #RMCS1,(R2)+
 1876 025030  112712  000200                MOVB    #200,(R2)
 1877
 1878 025034  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      025040  000404                        BR      250$            ;GO TO 250$ IF NO ERROR
      025042  000240                        NOP                     ;RETURN HERE IF ERROR
      025044  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      025046  000137  025420                JMP     350$            ;GO TO 350$ IF ERROR
 1879 025052                        250$:
 1880
 1881                               ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      025052  004737  044144                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 1882
 1883                               ;WAIT FOR COMMAND TO COMPLETE
      025056  004737  045042                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 1884
 1885                               ;GO READ STATUS FOR READ OPERATION
 1886 025062  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      025066  000404                        BR      260$            ;GO TO 260$ IF NO ERROR
      025070  000240                        NOP                     ;RETURN HERE IF ERROR
      025072  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      025074  000137  025420                JMP     350$            ;GO TO 350$ IF ERROR
 1887 025100                        260$:
 1888
 1889                               ;CHECK FOR ERRORS DURING READ OPERATION
 1890 025100  004737  045226                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      025104  000405                        BR      265$            ;GO TO 265$ IF NO ERROR
      025106  000240                        NOP                     ;RETURN HERE IF ERROR
      025110  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      025112  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      025114  000137  025420                JMP     350$            ;GO TO 350$ IF ERROR
 1891 025120                        265$:
 1892 025120  005704                        TST     R4              ;IS THIS THE FIRST HEADER WORD ?
 1893 025122  001006                        BNE     266$            ;NO !!
 1894
 1895 025124  032703  010000                BIT     #FMT16,R3       ;SHOULD FER BE SET ?
 1896 0 5130  001037                        BNE     275$            ;YES !!
 1897 025132  032703  140000                BIT     #MSE!USE,R3     ;SHOULD BSE BE SET ?
 1898 025136  001061                        BNE     285$            ;YES !!
 1899
 1900                               ;VERIFY THAT A HEADER COMPARE ERROR WAS DETECTED
 1901 025140                        266$:
 1902 025140  032737  000200  001350        BIT     #HCE,RMER1I     ;WAS HCE DETECTED ??
 1903 025146  001102                        BNE     295$            ;YES !!
 1904
 1905 025150  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      025154  000405                        BR      270$            ;GO TO 270$ IF NO ERROR
      025156  000240                        NOP                     ;RETURN HERE IF ERROR
      025160  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      025162  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      025164  000137  025420                JMP     350$            ;GO TO 350$ IF ERROR
 1906 025170                        270$:
 1907 025170  013737  001350  001142        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

```
1908 025176  012737  000200  001140          MOV     #HCE,$GDDAT     ;EXPECTED STATUS
1909 025204  010437  001174                  MOV     R4,$TMP0        ;R4 = HEADER WORD NUMBER AND
1910 025210  001002                          BNE     271$            ;ADJUST NUMBER.
1911 025212  005237  001174                  INC     $TMP0
1912 025216  010337  001176          271$:   MOV     R3,$TMP1        ;R3 = BIT POSITION
1913 025222  104344                          EMT     344
1914 025224  000137  025420                  JMP     350$
1915
1916                                 ;VERIFY THAT A FORMAT ERROR WAS DETECTED
1917 025230                          275$:
1918 025230  032737  000020  001350          BIT     #FER,RMER1I     ;WAS FER DETECTED ??
1919 025236  001046                          BNE     295$            ;YES !!
1920
1921 025240  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     025244  000405                          BR      280$            ;GO TO 280$ IF NO ERROR
     025246  000240                          NOP                     ;RETURN HERE IF ERROR
     025250  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     025252  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025254  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
1922 025260                          280$:
1923 025260  012737  000020  001140          MOV     #FER,$GDDAT     ;EXPECTED STATUS
1924 025266  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1925 025274  104343                          EMT     343
1926 025276  000137  025420                  JMP     350$
1927
1928                                 ;VERIFY THAT A BAD SECTOR ERROR WAS DETECTED
1929 025302                          285$:
1930 025302  032737  100000  001376          BIT     #BSE,RMER2I     ;WAS BSE DETECTED ??
1931 025310  001021                          BNE     295$            ;YES !!
1932
1933 025312  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     025316  000405                          BR      290$            ;GO TO 290$ IF NO ERROR
     025320  000240                          NOP                     ;RETURN HERE IF ERROR
     025322  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     025324  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025326  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
1934 025332                          290$:
1935 025332  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
1936 025340  012737  100000  001140          MOV     #BSE,$GDDAT     ;EXPECTED STAIUS
1937 025346  104345                          EMT     345
1938 025350  000137  000350                  JMP     350
1939
1940                                 ;CHECK FOR OTHER ERRORS
1941 025354                          295$:
1942 025354  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     025360  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
     025362  000240                          NOP                     ;RETURN HERE IF ERROR
     025364  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     025366  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025370  000137  025420                  JMP     350$            ;GO TO 350$ IF ERROR
1943 025374                          300$:
1944 025374  006303                          ASL     R3              ;SHIFT TO NEXT BIT
1945 025376  001006                          BNE     310$            ;REPEAT IF NOT DONE
1946
1947 025400  005704                          TST     R4              ;SECOND HEADER WORD DONE??
1948 025402  001006                          BNE     350$            ;YES!!
1949 025404  012704  000002                  MOV     #2,R4           ;SETUP FOR 2ND HEADER WORD
```

```
 1950 025410  012703  000001                MOV     #1,R3           ;SETUP FOR HCE
 1951 025414                        310$:
 1952 025414  000137  023762                JMP     15$
 1953 025420                        350$:
 1954 025420  000464                BR      370$            ;EXIT IF ERROR
 1955                               ;;*****************************************************************
 1956                               ;*REFORMAT SECTOR THAT WAS WRITTEN WITH BAD HEADER
 1957                               ;;*****************************************************************
 1958 025422                        355$:
 1959 025422  012737  177776  001412        MOV     #-2,RMWCO               ;ONLY TWO HEAD WORDS
 1960 025430  012737  010000  001442        MOV     #FMT16,RMOFO            ;ALWAYS IN 16 BITS MODE
 1961 025436  012737  106314  001414        MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS,REFORMAT THE SECTOR
 1962 025444  012737  000062  001410        MOV     #WH,RMCS1O              ;WRITE HEAD AND DATA COMMAND
 1963 025452  004737  043344                JSR     PC,GENBUF               ;SET UP THE BUFFER
 1964
 1965 025456  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      025462  054130                        .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                    ;VERIFY RECALIBRATION
      025464  000404                        BR      360$            ;GO TO 360$ IF NO ERROR
      025466  000240                        NOP                     ;RETURN HERE IF ERROR
      025470  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      025472  000137  025476                JMP     360$            ;GO TO 360$ IF ERROR
 1966 025476                        360$:
 1967 025476  012737  000063  001410        MOV     #WH!GO,RMCS1O           ;FORMAT THE SECTOR
 1968 025504  012702  001551                MOV     #PUTINX,R2              ;SET UP THE REGS
 1969 025510  112722  000006                MOVB    #RMDA,(R2)+
 1970 025514  112722  000032                MOVB    #RMOF,(R2)+
 1971 025520  112722  000034                MOVB    #RMDC,(R2)+
 1972 025524  112722  000004                MOVB    #RMBA,(R2)+
 1973 025530  112722  000002                MOVB    #RMWC,(R2)+
 1974 025534  112722  000000                MOVB    #RMCS1,(R2)+
 1975 025540  112722  000200                MOVB    #200,(R2)+
 1976 025544  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      025550  000404                        BR      365$            ;GO TO 365$ IF NO ERROR
      025552  000240                        NOP                     ;RETURN HERE IF ERROR
      025554  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      025556  000137  025572                JMP     370$            ;GO TO 370$ IF ERROR
 1977 025562  000240                365$:   NOP
 1978 025564  004737  045042                JSR     PC,TIMOUT               ;WAIT FOR FINISH
 1979 025570  000240                        NOP
 1980 025572                        370$:
 1981
 1982                               ;;*****************************************************************
                                    ;*TEST 17        WRITE, READ W/ HCI
                                    ;;*****************************************************************
      025572                        TST17:
      025572  000004                        SCOPE                   ;SCOPE CALL
      025574  000240                        NOP                     ;START OF TEST
      025576  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      025602  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      025606  013701  001464                MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      025612  012737  000017  001226        MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
```

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

025620  012704  000000          10$:    MOV     #0,R4           ;HEADER WORD
025624  012703  000001                  MOV     #1,R3           ;HCE BIT
025630                          15$:

                                ;PREPARE THE DEVICE FOR FORMAT OPERATION
025630  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
025634  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
025636  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
025640  000240                          NOP                     ;RETURN HERE IF ERROR
025642  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
025644  000137  027176                  JMP     370$            ;GO TO 370$ IF ERROR

                                ;LOAD PARAMETERS AND GENERATE DATA BUFFER
025650                          20$:
025650  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
025656  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
025664  012737  106314  001414          MOV     #BUFONE,RMBAO   ;BUS ADDRESS
025672  012737  177376  001412          MOV     #-258.,RMWCO    ;2 + 256. WORDS (2' COMP)
025700  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
025706  012737  000063  001410          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA COMMAND

                                ;VERIFY THAT SECTOR IS NOT BAD
025714  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
025720  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
025722  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
025726  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
025730  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
025734                          25$:
025734  012737  071666  001174          MOV     #ONES,$TMP0     ;STARTING ADDRESS OF PATTERN
025742  012737  000001  001176          MOV     #1,$TMP1        ;RANGE OF PATTERN
025750  004737  043344                  JSR     PC,GENBUF       ;GO GENERATE BUFFER FOR FORMAT

                                ;CHANGE HEADER WORD TO FORCE ERROR DURING WRITE
025754  030364  106314                  BIT     R3,BUFONE(R4)   ;SET OR RESET BIT??
025760  001403                          BEQ     27$
025762  040364  106314                  BIC     R3,BUFONE(R4)   ;RESET BIT
025766  000402                          BR      28$
025770  050364  106314          27$:    BIS     R3,BUFONE(R4)   ;SET BIT

                                ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
025774                          28$:
025774  012702  001551                  MOV     #PUTINX,R2      ;R2 - BYTE ENTRY POSITION
026000  112722  000034                  MOVB    #RMDC,(R2)+
026004  112722  000006                  MOVB    #RMDA,(R2)+
026010  112722  000004                  MOVB    #RMBA,(R2)+
026014  112722  000002                  MOVB    #RMWC,(R2)+
```

```
 2023 026020  112722  000032                MOVB    #RMOF,(R2)+
 2024 026024  112722  000000                MOVB    #RMCS1,(R2)+
 2025 026030  112712  000200                MOVB    #200,(R2)       ;TERMINATE TABLE
 2026 026034                          30$:
 2027
 2028                                 ;FORMAT THE DRIVE
 2029 026034  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      026040  000404                        BR      40$             ;GO TO 40$ IF NO ERROR
      026042  000240                        NOP                     ;RETURN HERE IF ERROR
      026044  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      026046  000137  027024                JMP     350$            ;GO TO 350$ IF ERROR
 2030 026052                          40$:
 2031
 2032                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      026052  004737  044144                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2033
 2034                                 ;WAIT FOR COMMAND TO COMPLETE
      026056  004737  045042                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2035
 2036
 2037 026062  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      026066  000404                        BR      50$             ;GO TC 50$ IF NO ERROR
      026070  000240                        NOP                     ;RETURN HERE IF ERROR
      026072  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      026074  000137  027024                JMP     350$            ;GO TO 350$ IF ERROR
 2038 026100                          50$:
 2039
 2040                                 ;VERIFY NO ERRORS DURING FORMAT
 2041 026100  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      026104  000405                        BR      60$             ;GO TO 60$ IF NO ERROR
      026106  000240                        NOP                     ;RETURN HERE IF ERROR
      026110  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      026112  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026114  000137  027024                JMP     350$            ;GO TO 350$ IF ERROR
 2042 026120                          60$:
 2043
 2044                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2045 026120  012737  026130  001124        MOV     #70$,$LPERR
 2046 026126  000410                        BR      80$             ;SKIP TO WRITE OPERATION
 2047
 2048                                 ;;*****************************************************************
 2049                                 ;LOOP #2        WRITE,READ
 2050
 2051 026130                          70$:
 2052
 2053                                 ;PREPARE DEVICE FOR WRITE OPERATION
 2054 026130  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      026134  154130                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                                    ;VERIFY RECALIBRATION
      026136  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
      026140  000240                        NOP                     ;RETURN HERE IF ERROR
```

```
        026142  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        026144  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
  2055  026150                          80$:
  2056
  2057                                ;WRITE DATA TO THE DRIVE
  2058  026150                        120$:
  2059  026150  012737  012000  001442          MOV     #FMT16!HCI,RMOFO        ;INHIBIT HEADER COMPARE
  2060  026156  012737  000061  001410          MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
  2061  026164  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;LOAD STARTING BUFFER ADDRESS
  2062  026172  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  2063  026176  112722  000006                  MOVB    #RMDA,(R2)+
  2064  026202  112722  000034                  MOVB    #RMDC,(R2)+
  2065  026206  112722  000032                  MOVB    #RMOF,(R2)+
  2066  026212  112722  000004                  MOVB    #RMBA,(R2)+
  2067  026216  112722  000002                  MOVB    #RMWC,(R2)+
  2068  026222  112722  000000                  MOVB    #RMCS1,(R2)+
  2069  026226  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
  2070
  2071  026232  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        026236  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        026240  000240                          NOP                     ;RETURN HERE IF ERROR
        026242  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        026244  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
  2072  026250                          130$:
  2073
  2074                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        026250  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  2075
  2076                                ;WAIT FOR COMMAND TO COMPLETE
        026254  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  2077
  2078                                ;GO READ STATUS FOR WRITE COMMAND
  2079  026260  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        026264  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        026266  000240                          NOP                     ;RETURN HERE IF ERROR
        026270  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        026272  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
  2080  026276                          140$:
  2081
  2082                                ;CHECK FOR ERRORS DURING WRITE OPERATION
  2083  026276  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        026302  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        026304  000240                          NOP                     ;RETURN HERE IF ERROR
        026306  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        026310  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        026312  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
  2084  026316  005037  001140          150$:   CLR     $GDDAT                  ;EXPECTED STATUS
  2085
  2086  026322  032737  000220  001350          BIT     #HCE!FER,RMER1I         ;ANY ERROR??
  2087  026330  001407                          BEQ     151$
  2088  026332  013737  001350  001142          MOV     RMER1I,$BDDAT           ;RECEIVED STATUS
  2089  026340  042737  177557  001142          BIC     #^C<HCE!FER>,$BDDAT     ;CLEAR DONT CARES
  2090  026346  000412                          BR      152$
  2091  026350                          151$:
  2092  026350  013737  001376  001142          MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
  2093  026356  032737  100000  001376          BIT     #BSE,RMER2I             ;ANY BAD SECTOR ERROR ?
  2094  026364  001406                          BEQ     155$                    ;NO !!
```

```
 2095 026366  042737  077777  001142          BIC     #^CBSE,$BDDAT           ;SAVE BSE FOR ERROR
 2096 026374                           152$:
 2097 026374  104346                          EMT     346
 2098 026376  000137  027024                  JMP     350$
 2099 026402                           155$:
 2100
 2101 026402                           160$:
 2102 026402  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      026406  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
      026410  000240                          NOP                     ;RETURN HERE IF ERROR
      026412  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      026414  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026416  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2103 026422                           170$:
 2104 026422  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      026426  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
      026430  000240                          NOP                     ;RETURN HERE IF ERROR
      026432  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      026434  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026436  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2105 026442                           180$:
 2106
 2107                                 ;CHANGE LOOP ADDRESSES
 2108 026442  012737  026452  001124          MOV     #190$,$LPERR
 2109 026450  000410                          BR      200$
 2110
 2111                                 ;;*****************************************************************
 2112                                 ;LOOP #3        READ
 2113
 2114 026452                           190$:
 2115
 2116                                 ;PREPARE DEVICE FOR READ OPERATION
 2117 026452  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      026456  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                               ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                               ;VERIFY PACK ACKNOWLEDGE
                                               ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                               ;VERIFY RECALIBRATION
      026460  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
      026462  000240                          NOP                     ;RETURN HERE IF ERROR
      026464  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      026466  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2118 026472                           200$:
 2119
 2120                                 ;READ DATA FROM DEVICE
 2121 026472                           240$:
 2122 026472  012737  000071  001410          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
 2123 026500  012737  107324  001414          MOV     #BUFTWO+4,RMBA0         ;LOAD STARTING BUFFER ADDRESS
 2124 026506  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2125 026512  112722  000006                  MOVB    #RMDA,(R2)+
 2126 026516  112722  000032                  MOVB    #RMOF,(R2)+
 2127 026522  112722  000034                  MOVB    #RMDC,(R2)+
 2128 026526  112722  000004                  MOVB    #RMBA,(R2)+
 2129 026532  112722  000002                  MOVB    #RMWC,(R2)+
```

```
 2130 026536  112722  000000                  MOVB    #RMCS1,(R2)+
 2131 026542  112712  000200                  MOVB    #200,(R2)
 2132
 2133 026546  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      026552  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
      026554  000240                          NOP                     ;RETURN HERE IF ERROR
      026556  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      026560  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2134 026564                          250$:
 2135
 2136                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      026564  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2137
 2138                                 ;WAIT FOR COMMAND TO COMPLETE
      026570  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2139
 2140                                 ;GO READ STATUS FOR READ OPERATION
 2141 026574  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      026600  000404                          BR      260$            ;GO TO 260$ IF NO ERROR
      026602  000240                          NOP                     ;RETURN HERE IF ERROR
      026604  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      026606  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2142 026612                          260$:
 2143
 2144                                 ;CHECK FOR ERRORS DURING READ OPERATION
 2145 026612  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      026616  000405                          BR      270$            ;GO TO 270$ IF NO ERROR
      026620  000240                          NOP                     ;RETURN HERE IF ERROR
      026622  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      026624  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026626  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2146 026632  005037  001140          270$:   CLR     $GDDAT                  ;EXPECTED STATUS
 2147
 2148 026636  032737  000220  001350          BIT     #HCE!FER,RMER1I         ;ANY ERROR ?
 2149 026644  001407                          BEQ     271$            ;NO!!
 2150 026646  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 2151 026654  042737  177557  001142          BIC     #^C<HCE!FER>,$BDDAT     ;CLEAR DONT CARES
 2152 026662  000412                          BR      272$
 2153 026664                          271$:
 2154 026664  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
 2155 026672  032737  100000  001376          BIT     #BSE,RMER2I     ;ANY BAD SECTOR ERROR ?
 2156 026700  001406                          BEQ     275$            ;NO !!
 2157 026702  042737  077777  001142          BIC     #^CBSE,$BDDAT   ;CLEAR DONT CARES
 2158 026710                          272$:
 2159 026710  104346                          EMT     346
 2160 026712  000137  027024                  JMP     350$
 2161 026716                          275$:
 2162
 2163 026716                          280$:
 2164 026716  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      026722  000405                          BR      290$            ;GO TO 290$ IF NO ERROR
      026724  000240                          NOP                     ;RETURN HERE IF ERROR
      026726  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      026730  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026732  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
 2165 026736                          290$:
 2166 026736  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
```

```
        026742  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
        026744  000240                          NOP                     ;RETURN HERE IF ERROR
        026746  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        026750  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        026752  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
  2167  026756                          300$:
  2168  026756  004737  043602                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        026762  106320                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        026764  107324                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        026766  000404                          BR      310$            ;GO TO 310$ IF NO ERROR
        026770  000240                          NOP                     ;RETURN HERE IF ERROR
        026772  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        026774  000137  027024                  JMP     350$            ;GO TO 350$ IF ERROR
  2169  027000                          310$:
  2170  027000  006303                          ASL     R3                      ;SHIFT HCE BIT
  2171  027002  001006                          BNE     320$                    ;CONTINUE IF NOT DONE
  2172  027004  005704                          TST     R4                      ;SECOND HEADER DONE??
  2173  027006  001006                          BNE     350$                    ;YES!!
  2174  027010  012703  000001                  MOV     #1,R3                   ;START WITH BIT 0
  2175  027014  012704  000002                  MOV     #2,R4                   ;DO SECOND HEADER WORD
  2176  027020                          320$:
  2177  027020  000137  025630                  JMP     15$
  2178  027024                          350$:
  2179  027024  000464                          BR      370$            ;EXIT IF ERROR
  2180                                  ;;*****************************************************************
  2181                                  ;*REFORMAT SECTOR THAT WAS WRITTEN WITH BAD HEADER
  2182                                  ;;*****************************************************************
  2183  027026                          355$:
  2184  027026  012737  177776  001412          MOV     #-2,RMWCO               ;WORD COUNT
  2185  027034  012737  010000  001442          MOV     #FMT16,RMOFO            ;IN 16 BIT MODE
  2186  027042  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS
  2187  027050  012737  000062  001410          MOV     #WH,RMCS1O              ;FORMAT COMMAND
  2188  027056  004737  043344                  JSR     PC,GENBUF               ;SET UP BUFFER
  2189
  2190  027062  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        027066  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        027070  000404                          BR      360$            ;GO TO 360$ IF NO ERROR
        027072  000240                          NOP                     ;RETURN HERE IF ERROR
        027074  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        027076  000137  027102                  JMP     360$            ;GO TO 360$ IF ERROR
  2191  027102                          360$:
  2192  027102  012737  000063  001410          MOV     #WH!GO,RMCS1O
  2193  027110  012702  001551                  MOV     #PUTINX,R2              ;TABLE ADDRESS
  2194  027114  112722  000006                  MOVB    #RMDA,(R2)+
  2195  027120  112722  000032                  MOVB    #RMOF,(R2)+
  2196  027124  112722  000034                  MOVB    #RMDC,(R2)+
  2197  027130  112722  000004                  MOVB    #RMBA,(R2)+
  2198  027134  112722  000002                  MOVB    #RMWC,(R2)+
  2199  027140  112722  000000                  MOVB    #RMCS1,(R2)+
  2200  027144  112722  000200                  MOVB    #200,(R2)+
  2201  027150  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
```

```
                027154  000404                          BR      365$            ;GO TO 365$ IF NO ERROR
                027156  000240                          NOP                     ;RETURN HERE IF ERROR
                027160  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                027162  000137  027176                  JMP     370$            ;GO TO 370$ IF ERROR
     2202       027166  000240                  365$:   NOP
     2203       027170  004737  045042                  JSR     PC,TIMOUT
     2204       027174  000240                          NOP
     2205       027176                          370$:
     2206
     2207                                       ;;*****************************************************************
                                                ;*TEST 20        WRITE, READ W/ IVC ERROR
                                                ;;*****************************************************************
                027176                          TST20:
                027176  000004                          SCOPE                   ;SCOPE CALL
                027200  000240                          NOP                     ;START OF TEST
                027202  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                027206  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                027212  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                027216  012737  000020  001226          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     2208
     2209                                       ;;*****************************************************************
     2210                                       ;LOOP #1        WRITE,READ
     2211
     2212                                       ;LOAD PARAMETERS AND GENERATE DATA BUFFER
     2213       027224                          10$:
     2214       027224  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
     2215       027232  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
     2216       027240  012737  106314  001414          MOV     #BUFONE,RMBA0   ;BUS ADDRESS
     2217       027246  012737  177400  001412          MOV     #-256.,RMWC0    ;256. WORDS (2'S COMP)
     2218       027254  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
     2219       027262  012737  000060  001410          MOV     #WD,RMCS10      ;WRITE HEADER AND DATA COMMAND
     2220
     2221                                       ;VERIFY THAT SECTOR IS NOT BAD
                027270  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
                027274  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
                027276  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
                027302  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
                027304  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
     2222       027310                          70$:
     2223       027310  012737  001416  001174          MOV     #RMDA0,$TMP0    ;USE SECTOR FOR DATA PATTERN
     2224       027316  012737  000001  001176          MOV     #1,$TMP1
     2225       027324  004737  043344                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
     2226
     2227                                       ;PREPARE DEVICE FOR WRITE OPERATION
     2228       027330  004737  040472          JSR     PC,TSTPRP               ;PREPARE DEVICE FOR TEST
                027334  154130                  .WORD   154130                  ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ,PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                027336  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                027340  000240                          NOP                     ;RETURN HERE IF ERROR
                027342  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                027344  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
```

```
 2229 027350                          80$:
 2230
 2231                                 ;RESET VOLUME VALID
 2232 027350  112737  000024  001551          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
      027356  112737  000200  001552          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      027364  012737  000001  001434          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
      027372  004737  044500                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
      027376  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
      027400  000240                          NOP                     ;RETURN HERE IF ERROR
      027402  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      027404  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2233 027410                          90$:
 2234
 2235                                 ;SETUP PARAMETERS AND EXCUTE WRITE DATA COMMAND
 2236 027410                          120$:
 2237 027410  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2238 027416  012737  000000  001434          MOV     #0,RMMR10               ;RESET DIAGNOSTIC MODE
 2239 027424  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2240 027430  112722  000024                  MOVB    #RMMR1,(R2)+
 2241 027434  112722  000006                  MOVB    #RMDA,(R2)+
 2242 027440  112722  000034                  MOVB    #RMDC,(R2)+
 2243 027444  112722  000032                  MOVB    #RMOF,(R2)+
 2244 027450  112722  000002                  MOVB    #RMWC,(R2)+
 2245 027454  112722  000004                  MOVB    #RMBA,(R2)+
 2246 027460  112722  000000                  MOVB    #RMCS1,(R2)+
 2247 027464  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 2248
 2249 027470  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      027474  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
      027476  000240                          NOP                     ;RETURN HERE IF ERROR
      027500  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      027502  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2250 027506                          130$:
 2251
 2252                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      027506  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2253
 2254                                 ;WAIT FOR COMMAND TO COMPLETE
      027512  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2255
 2256                                 ;GO READ STATUS FOR WRITE COMMAND
 2257 027516  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      027522  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
      027524  000240                          NOP                     ;RETURN HERE IF ERROR
      027526  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      027530  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2258 027534                          140$:
 2259
 2260                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
 2261 027534  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      027540  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
      027542  000240                          NOP                     ;RETURN HERE IF ERROR
      027544  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      027546  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      027550  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2262 027554                          150$:
 2263 027554  032737  010000  001376          BIT     #IVC,RMER2I             ;WAS ''IVC'' DETECTED??
```

```
 2264 027562  001024                          BNE     170$                    ;YES!!
 2265
 2266 027564  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      027570  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
      027572  000240                          NOP                     ;RETURN HERE IF ERROR
      027574  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      027576  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      027600  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2267 027604                          160$:
 2268 027604  013737  001376  001142          MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
 2269 027612  013737  001376  001140          MOV     RMER2I,$GDDAT           ;EXPECTED STATUS
 2270 027620  052737  010000  001140          BIS     #IVC,$GDDAT
 2271 027626  104342                          EMT     342
 2272 027630  000137  030216                  JMP     350$
 2273 027634                          170$:
 2274 027634  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      027640  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
      027642  000240                          NOP                     ;RETURN HERE IF ERROR
      027644  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      027646  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      027650  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2275 027654                          180$:
 2276
 2277                                 ;CHANGE LOOP ADDRESSES
 2278 027654  012737  027664  001124          MOV     #190$,$LPERR
 2279 027662  000410                          BR      200$
 2280
 2281                                 ;;*****************************************************************
 2282                                 ;LOOP #2        READ
 2283
 2284 027664                          190$:
 2285
 2286                                 ;PREPARE DEVICE FOR READ OPERATION
 2287 027664  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      027670  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                               ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                               ;VERIFY PACK ACKNOWLEDGE
                                               ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                               ;VERIFY RECALIBRATION
      027672  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
      027674  000240                          NOP                     ;RETURN HERE IF ERROR
      027676  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      027700  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2288 027704                          200$:
 2289
 2290                                 ;RESET VOLUME VALID
 2291 027704  112737  000024  001551          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
      027712  112737  000200  001552          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      027720  012737  000001  001434          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
      027726  004737  044500                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
      027732  000404                          BR      210$            ;GO TO 210$ IF NO ERROR
      027734  000240                          NOP                     ;RETURN HERE IF ERROR
      027736  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      027740  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
```

```
                                        210$:

                                        ;READ DATA FROM DEVICE
                                        240$:
001410                                          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
001414                                          MOV     #BUFTWO,RMBA0           ;LOAD STARTING BUFFER ADDRESS
001434                                          MOV     #0,RMMR10               ;RESET DIANOSTIC MODE
                                                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
                                                MOVB    #RMMR1,(R2)+
                                                MOVB    #RMDA,(R2)+
                                                MOVB    #RMOF,(R2)+
                                                MOVB    #RMDC,(R2)+
                                                MOVB    #RMBA,(R2)+
                                                MOVB    #RMWC,(R2)+
                                                MOVB    #RMCS1,(R2)+
                                                MOVB    #200,(R2)

030216                                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                                BR      250$            ;GO TO 250$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                                JMP     350$            ;GO TO 350$ IF ERROR
                                        250$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
044144                                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
045042                                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR READ OPERATION
044230                                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                BR      260$            ;GO TO 260$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
030216                                          JMP     350$            ;GO TO 350$ IF ERROR
                                        260$:

                                        ;CHECK FOR ERRORS DURING READ OPERATION
045226                                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                                                BR      270$            ;GO TO 270$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030216                                          JMP     350$            ;GO TO 350$ IF ERROR
                                        270$:
001376                                          BIT     #IVC,RMER2I             ;WAS ''IVC'' DETECTED??
                                                BNE     290$                    ;YES!!

057742                                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                                BR      280$            ;GO TO 280$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030216                                          JMP     350$            ;GO TO 350$ IF ERROR
                                        280$:
001376  001140                                 MOV     RMER2I,$GDDAT           ;EXPECTED STATUS
```

```
 2329 030154  052737  010000  001140          BIS     #IVC,$GDDAT
 2330 030162  013737  001376  001142          MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
 2331 030170  104342                          EMT     342
 2332 030172  000137  030216                  JMP     350$
 2333 030176                          290$:
 2334 030176  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      030202  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
      030204  000240                          NOP                     ;RETURN HERE IF ERROR
      030206  104000                          EMT                     ;ERROR # DFFINED BY SECERR SUBROUTINE
      030210  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      030212  000137  030216                  JMP     350$            ;GO TO 350$ IF ERROR
 2335 030216                          300$:
 2336
 2337 030216                          350$:
 2338
 2339                                 ;;*****************************************************************
                                      ;*TEST 21        WRITE, READ W/ ABORT
                                      ;;*****************************************************************
      030216                          TST21:
      030216  000004                          SCOPE                   ;SCOPE CALL
      030220  000240                          NOP                     ;START OF TEST
      030222  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      030226  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      030232  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      030236  012737  000021  001226          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2340
 2341                                 ;;*****************************************************************
 2342                                 ;LOOP #1        WRITE,READ
 2343
 2344                                 ;LOAD PARAMETERS AND GENERATE DATA BUFFER
 2345 030244  012737  000000  001444          MOV     #0,RMDC0                ;CYLINDER = 0
 2346 030252  012737  000000  001416          MOV     #0,RMDA0                ;TRACK = 0, SECTOR = 0
 2347 030260  012737  106314  001414          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
 2348 030266  012737  177400  001412          MOV     #-256.,RMWC0            ;256. WORDS (2'S COMP)
 2349 030274  012737  010000  001442          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
 2350 030302  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2351 030310                          70$:
 2352
 2353                                 ;PREPARE DEVICE FOR WRITE OPERATION
 2354 030310  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      030314  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                              ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                              ;CLEAR CONTROLLER & SELECT DEVICE
                                              ;VERIFY CONTROLLER CLEAR OPERATION
                                              ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                              ;VERIFY PACK ACKNOWLEDGE
                                              ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                              ;VERIFY RECALIBRATION
      030316  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
      030320  000240                          NOP                     ;RETURN HERE IF ERROR
      030322  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      030324  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
 2355 030330                          80$:
 2356
 2357                                 ;SET UNSAFE ERROR
 2358 030330  112737  000014  001551          MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
      030336  112737  000200  001552          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
```

```
        030344  012737  040000  001424          MOV     #UNS,RMER10     ;SET RMER1 OUTPUT BUFFER = UNS
        030352  004737  044500                  JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
        030356  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        030360  000240                          NOP                     ;RETURN HERE IF ERROR
        030362  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        030364  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
 2359   030370                          90$:
 2360
 2361                                   ;WRITE DATA TO THE DRIVE
 2362   030370                          120$:
 2363   030370  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2364   030376  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2365   030402  112722  000006                  MOVB    #RMDA,(R2)+
 2366   030406  112722  000034                  MOVB    #RMDC,(R2)+
 2367   030412  112722  000032                  MOVB    #RMOF,(R2)+
 2368   030416  112722  000004                  MOVB    #RMBA,(R2)+
 2369   030422  112722  000002                  MOVB    #RMWC,(R2)+
 2370   030426  112722  000000                  MOVB    #RMCS1,(R2)+
 2371   030432  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 2372
 2373   030436  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        030442  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        030444  000240                          NOP                     ;RETURN HERE IF ERROR
        030446  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        030450  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
 2374   030454                          130$:
 2375
 2376                                   ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        030454  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2377
 2378                                   ;GO GET REGISTER STATUS
 2379   030460  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        030464  000404                          BR      135$            ;GO TO 135$ IF NO ERROR
        030466  000240                          NOP                     ;RETURN HERE IF ERROR
        030470  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        030472  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
 2380   030476                          135$:
 2381   030476  032737  020000  001346          BIT     #PIP,RMDSI              ;WAS COMMAND EXECUTED?
 2382   030504  001412                          BEQ     136$                    ;NO, GO WAIT
 2383   030506  013737  001346  001140          MOV     RMDSI,$GDDAT            ;EXPECTED STATUS
 2384   030514  042737  020000  001140          BIC     #PIP,$GDDAT
 2385   030522  013737  001346  001142          MOV     RMDSI,$BDDAT            ;RECEIVED STATUS
 2386   030530  104347                          EMT     347
 2387   030532                          136$:
 2388
 2389                                   ;WAIT FOR COMMAND TO COMPLETE
        030532  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2390
 2391                                   ;GO GET REGISTER STATUS
 2392   030536  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        030542  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
        030544  000240                          NOP                     ;RETURN HERE IF ERROR
        030546  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        030550  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
 2393   030554                          140$:
 2394   030554  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        030560  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
```

```
        030562  000240                          NOP                     ;RETURN HERE IF ERROR
        030564  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        030566  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030570  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
  2395  030574                          150$:
  2396  030574  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        030600  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
        030602  000240                          NOP                     ;RETURN HERE IF ERROR
        030604  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        030606  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030610  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
  2397  030614                          180$:
  2398
  2399                                  ;CHANGE LOOP ADDRESSES
  2400  030614  012737  030624  001124          MOV     #190$,$LPERR
  2401  030622  000410                          BR      200$
  2402
  2403                                  ;;**********************************************************************
  2404                                  ;LOOP #2        READ
  2405
  2406  030624                          190$:
  2407
  2408                                  ;PREPARE DEVICE FOR READ OPERATION
  2409  030624  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        030630  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        030632  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
        030634  000240                          NOP                     ;RETURN HERE IF ERROR
        030636  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        030640  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
  2410  030644                          200$:
  2411
  2412                                  ;SET UNSAFE ERROR
  2413  030644  112737  000014  001551          MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
        030652  112737  000200  001552          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        030660  012737  040000  001424          MOV     #UNS,RMFR10     ;SET RMER1 OUTPUT BUFFER = UNS
        030666  004737  044500                  JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
        030672  000404                          BR      210$            ;GO TO 210$ IF NO ERROR
        030674  000240                          NOP                     ;RETURN HERE IF ERROR
        030676  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        030700  000137  031134                  JMP     350$            ;GO TO 350$ IF ERROR
  2414  030704                          210$:
  2415
  2416                                  ;READ DATA FROM DEVICE
  2417  030704                          240$:
  2418  030704  012737  000071  001410          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
  2419  030712  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  2420  030716  112722  000006                  MOVB    #RMDA,(R2)+
  2421  030722  112722  000032                  MOVB    #RMOF,(R2)+
  2422  030726  112722  000034                  MOVB    #RMDC,(R2)+
  2423  030732  112722  000004                  MOVB    #RMBA,(R2)+
```

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

                                                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                                BR      250$            ;GO TO 250$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                                JMP     350$            ;GO TO 350$ IF ERROR
                                        250$:

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

                                        ;SEE IF DEVICE STARTED COMMAND
                                                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                BR      255$            ;GO TO 255$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                                JMP     350$            ;GO TO 350$ IF ERROR
                                        255$:
                                                BIT     #PIP,RMDSI              ;WAS COMMAND EXECUTED??
                                                BEQ     256$                    ;NO!!
                                                MOV     RMDSI,$GDDAT            ;EXPECTED STATUS
                                                BIC     #PIP,$GDDAT
                                                MOV     RMDSI,$BDDAT            ;RECEIVED STATUS
                                                EMT     347
                                                JMP     350$
                                        256$:

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

                                        ;GO READ STATUS FOR READ OPERATION
                                                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                                BR      260$            ;GO TO 260$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                                JMP     350$            ;GO TO 350$ IF ERROR
                                        260$:
                                                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                                                BR      270$            ;GO TO 270$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                                JMP     350$            ;GO TO 350$ IF ERROR
                                        270$:
                                                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                                                BR      300$            ;GO TO 300$ IF NO ERROR
                                                NOP                     ;RETURN HERE IF ERROR
                                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                                                JSR     PC,@(SP)+       ;CO BACK FOR MORE ERROR CHECKS
                                                JMP     350$            ;GO TO 350$ IF ERROR
                                        300$:

                                        350$:
```

```
2457                                    ;;*****************************************************************
                                        ;*TEST 22       WRITE, READ EACH CURRENT LEVEL
                                        ;;*****************************************************************
        031134                          TST22:
        031134  000004                          SCOPE                   ;SCOPE CALL
        031136  000240                          NOP                     ;START OF TEST
        031140  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        031144  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        031150  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        031154  012737  000022  001226          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2458    031162                          10$:
2459
2460                                    ;;*****************************************************************
2461                                    ;LOOP #1        FORMAT,WRITE,READ
2462
2463                                    ;PREPARE THE DEVICE FOR FORMAT OPERATION
2464    031162  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        031166  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
        031170  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        031172  000240                          NOP                     ;RETURN HERE IF ERROR
        031174  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        031176  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
2465    031202                          20$:
2466
2467                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
2468    031202  012737  000000  001444          MOV     #0,RMDCO                ;CYLINDER = 0
2469    031210  012737  000000  001416  25$:     MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
2470    031216  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
2471    031224  012737  177376  001412          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
2472    031232  012737  010000  001442          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
2473    031240  012737  000063  001410          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
2474
2475                                    ;VERIFY THAT SECTOR IS NOT BAD
        031246  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        031252  000405                          BR      26$             ;GO TO 26$ IF NO ERROR
        031254  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        031260  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        031262  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
2476    031266                          26$:
2477    031266  012737  071730  001174          MOV     #ZEROS,$TMPO            ;STARTING ADDRESS OF PATTERN
2478    031274  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
2479    031302  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
2480
2481                                    ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
2482    031306  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
2483    031312  112722  000034                  MOVB    #RMDC,(R2)+
2484    031316  112722  000006                  MOVB    #RMDA,(R2)+
2485    031322  112722  000004                  MOVB    #RMBA,(R2)+
2486    031326  112722  000002                  MOVB    #RMWC,(R2)+
2487    031332  112722  000032                  MOVB    #RMOF,(R2)+
2488    031336  112722  000000                  MOVB    #RMCS1,(R2)+
```

```
 2489 031342  112712  000200              MOVB    #200,(R2)               ;TERMINATE TABLE
 2490 031346                        30$:
 2491 031346  004737  044500              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      031352  000404                      BR      40$             ;GO TO 40$ IF NO ERROR
      031354  000240                      NOP                     ;RETURN HERE IF ERROR
      031356  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      031360  000137  032172              JMP     350$            ;GO TO 350$ IF ERROR
 2492 031364                        40$:
 2493
 2494                               ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      031364  004737  044144              JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2495
 2496                               ;WAIT FOR COMMAND TO COMPLETE
      031370  004737  045042              JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2497
 2498                               ;GO READ STATUS FOR FORMAT OPERATION
 2499 031374  004737  044230              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      031400  000404                      BR      50$             ;GO TO 50$ IF NO ERROR
      031402  000240                      NOP                     ;RETURN HERE IF ERROR
      031404  104000                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      031406  000137  032172              JMP     350$            ;GO TO 350$ IF ERROR
 2500 031412                        50$:
 2501
 2502                               ;VERIFY NO ERRORS DURING FORMAT
 2503 031412  004737  057742              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      031416  000405                      BR      60$             ;GO TO 60$ IF NO ERROR
      031420  000240                      NOP                     ;RETURN HERE IF ERROR
      031422  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      031424  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      031426  000137  032172              JMP     350$            ;GO TO 350$ IF ERROR
 2504 031432                        60$:
 2505
 2506                               ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2507 031432  012737  031442  001124        MOV     #70$,$LPERR
 2508 031440  000410                      BR      80$                     ;SKIP TO WRITE OPERATION
 2509
 2510                               ;;*****************************************************************
 2511                               ;LOOP #2        WRITE,READ
 2512
 2513 031442                        70$:
 2514
 2515
 2516                               ;PREPARE DEVICE FOR WRITE OPERATION
 2517 031442  004737  040472              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      031446  054130                      .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                   ;CLEAR CONTROLLER & SELECT DEVICE
                                                                   ;VERIFY CONTROLLER CLEAR OPERATION
                                                                   ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                   ;VERIFY PACK ACKNOWLEDGE
                                                                   ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                   ;VERIFY RECALIBRATION
      031450  000404                      BR      80$             ;GO TO 80$ IF NO ERROR
      031452  000240                      NOP                     ;RETURN HERE IF ERROR
      031454  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      031456  000137  032172              JMP     350$            ;GO TO 350$ IF ERROR
 2518 031462                        80$:
 2519
```

```
 2520 031462                          120$:
 2521
 2522                                 ;WRITE DATA TO THE DRIVE
 2523 031462  012737  177400  001412          MOV     #-256.,RMWCO            ;256. WORDS (2'S COMP)
 2524 031470  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
 2525 031476  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2526 031504  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2527 031510  112722  000006                  MOVB    #RMDA,(R2)+
 2528 031514  112722  000034                  MOVB    #RMDC,(R2)+
 2529 031520  112722  000032                  MOVB    #RMOF,(R2)+
 2530 031524  112722  000004                  MOVB    #RMBA,(R2)+
 2531 031530  112722  000002                  MOVB    #RMWC,(R2)+
 2532 031534  112722  000000                  MOVB    #RMCS1,(R2)+
 2533 031540  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 2534
 2535 031544  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      031550  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
      031552  000240                          NOP                     ;RETURN HERE IF ERROR
      031554  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      031556  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
 2536 031562                          130$:
 2537
 2538                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      031562  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2539
 2540                                 ;WAIT FOR COMMAND TO COMPLETE
      031566  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2541
 2542                                 ;GO READ STATUS FOR WRITE COMMAND
 2543 031572  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      031576  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
      031600  000240                          NOP                     ;RETURN HERE IF ERROR
      031602  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      031604  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
 2544 031610                          140$:
 2545
 2546                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
 2547 031610  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      031614  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
      031616  000240                          NOP                     ;RETURN HERE IF ERROR
      031620  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      031622  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      031624  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
 2548 031630                          150$:
 2549 031630  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      031634  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
      031636  000240                          NOP                     ;RETURN HERE IF ERROR
      031640  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      031642  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      031644  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
 2550 031650                          160$:
 2551
 2552 031650                          170$:
 2553 031650  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      031654  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
      031656  000240                          NOP                     ;RETURN HERE IF ERROR
      031660  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
```

```
        031662  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031664  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
  2554  031670                          180$:
  2555
  2556                                  ;CHANGE LOOP ADDRESSES
  2557  031670  012737  031700  001124          MOV     #190$,$LPERR
  2558  031676  000410                          BR      200$                    ;SKIP TO NEXT OPERATION
  2559
  2560                                  ;;*****************************************************************
  2561                                  ;LOOP #3        READ
  2562
  2563  031700                          190$:
  2564
  2565                                  ;PREPARE DEVICE FOR READ OPERATION
  2566  031700  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        031704  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        031706  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
        031710  000240                          NOP                     ;RETURN HERE IF ERROR
        031712  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        031714  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
  2567  031720                          200$:
  2568
  2569  031720                          240$:
  2570
  2571                                  ;READ DATA FROM DEVICE
  2572  031720  012737  107324  001414          MOV     #BUFTWO+4,RMBAO         ;CHANGE ADDRESS
  2573  031726  012737  177400  001412          MOV     #-256.,RMWCO            ;256. WORDS (2'S COMP)
  2574  031734  012737  000071  001410          MOV     #RD!GO,RMCS1O           ;READ DATA COMMAND
  2575  031742  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  2576  031746  112722  000006                  MOVB    #RMDA,(R2)+
  2577  031752  112722  000032                  MOVB    #RMOF,(R2)+
  2578  031756  112722  000034                  MOVB    #RMDC,(R2)+
  2579  031762  112722  000004                  MOVB    #RMBA,(R2)+
  2580  031766  112722  000002                  MOVB    #RMWC,(R2)+
  2581  031772  112722  000000                  MOVB    #RMCS1,(R2)+
  2582  031776  112712  000200                  MOVB    #200,(R2)
  2583
  2584  032002  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        032006  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
        032010  000240                          NOP                     ;RETURN HERE IF ERROR
        032012  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        032014  000137  032172                  JMP     350$            ;GO TO 350$ IF ERROR
  2585  032020                          250$:
  2586
  2587                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        032020  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  2588
  2589                                  ;WAIT FOR COMMAND TO COMPLETE
        032024  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  2590
  2591                                  ;GO READ STATUS FOR READ OPERATION
```

```
2592 032030 004737 044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032034 000404                 BR      260$            ;GO TO 260$ IF NO ERROR
     032036 000240                 NOP                     ;RETURN HERE IF ERROR
     032040 104000                 EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032042 000137 032172          JMP     350$            ;GO TO 350$ IF ERROR
2593 032046                    260$:
2594
2595                           ;CHECK FOR ERRORS DURING READ OPERATION
2596 032046 004737 045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     032052 000405                 BR      270$            ;GO TO 270$ IF NO ERROR
     032054 000240                 NOP                     ;RETURN HERE IF ERROR
     032056 104000                 EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     032060 004736                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032062 000137 032172          JMP     350$            ;GO TO 350$ IF ERROR
2597 032066                    270$:
2598 032066 004737 057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     032072 000405                 BR      280$            ;GO TO 280$ IF NO ERROR
     032074 000240                 NOP                     ;RETURN HERE IF ERROR
     032076 104000                 EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     032100 004736                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032102 000137 032172          JMP     350$            ;GO TO 350$ IF ERROR
2599 032106                    280$:
2600
2601 032106                    290$:
2602 032106 004737 046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     032112 000405                 BR      300$            ;GO TO 300$ IF NO ERROR
     032114 000240                 NOP                     ;RETURN HERE IF ERROR
     032116 104000                 EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     032120 004736                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032122 000137 032172          JMP     350$            ;GO TO 350$ IF ERROR
2603 032126                    300$:
2604 032126 004737 043602          JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
     032132 106320                 .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
     032134 107324                 .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
     032136 000404                 BR      310$            ;GO TO 310$ IF NO ERROR
     032140 000240                 NOP                     ;RETURN HERE IF ERROR
     032142 104000                 EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
     032144 000137 032172          JMP     350$            ;GO TO 350$ IF ERROR
2605 032150                    310$:
2606 032150 062737 000200 001444   ADD     #128.,RMDCO             ;ADVANCE TO NEXT THRESHOLD
2607 032156 023727 001444 001400   CMP     RMDCO,#768.             ;DONE??
2608 032164 101002                 BHI     350$                    ;YES!!
2609 032166 000137 031210          JMP     25$                     ;DO NEXT CURRENT LEVEL
2610 032172                    350$:
2611
2612                           ;;*****************************************************************
                               ;*TEST 23        WRITE, WRITE CHECK W/ CURRENT LEVEL SWITCHING
                               ;;*****************************************************************
     032172                    TST23:
     032172 000004                 SCOPE                   ;SCOPE CALL
     032174 000240                 NOP                     ;START OF TEST
     032176 012706 001100          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     032202 013700 001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     032206 013701 001464          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     032212 012737 000023 001226   MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2613 032220                    10$:
2614
```

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

032220  004737  040472          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
032224  054130                  .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                        ;VERIFY PACK ACKNOWLEDGE
                                                        ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                        ;VERIFY RECALIBRATION
032226  000404                  BR      20$             ;GO TO 20$ IF NO ERROR
032230  000240                  NOP                     ;RETURN HERE IF ERROR
032232  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
032234  000137  033176          JMP     350$            ;GO TO 350$ IF ERROR
032240                  20$:

                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
032240  012737  000577  001444          MOV     #383.,RMDC0             ;CYLINDER = 383.
032246  013737  001332  001416          MOV     LSTRK,RMDA0             ;SET LAST TRACK AND
032254  112737  000037  001416          MOVB    #31.,RMDA0              ;LAST SECTOR
032262  012737  106314  001414          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
032270  012737  176774  001412          MOV     #-258.*2,RMWC0          ;2 SECTORS (2'S COMP)
032276  012737  010000  001442          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
032304  012737  000063  001410          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND

                        ;VERIFY THAT SECTOR IS NOT BAD
032312  004737  041416          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
032316  000405                  BR      25$             ;GO TO 25$ IF NO ERROR
032320  104401  070230          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
032324  104000                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
032326  000137  033176          JMP     350$            ;GO TO 350$ IF ERROR
032332                  25$:
032332  012737  071666  001174          MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
032340  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
032346  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
032352  012702  001551          MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
032356  112722  000034          MOVB    #RMDC,(R2)+
032362  112722  000006          MOVB    #RMDA,(R2)+
032366  112722  000004          MOVB    #RMBA,(R2)+
032372  112722  000002          MOVB    #RMWC,(R2)+
032376  112722  000032          MOVB    #RMOF,(R2)+
032402  112722  000000          MOVB    #RMCS1,(R2)+
032406  112712  000200          MOVB    #200,(R2)               ;TERMINATE TABLE
032412                  30$:
032412  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
032416  000404                  BR      40$             ;GO TO 40$ IF NO ERROR
032420  000240                  NOP                     ;RETURN HERE IF ERROR
032422  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
032424  000137  033176          JMP     350$            ;GO TO 350$ IF ERROR
032430                  40$:

                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
032430  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
```

```
2651                                    ;WAIT FOR COMMAND TO COMPLETE
     032434  004737  045042                     JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2652
2653                                    ;GO READ STATUS FOR FORMAT OPERATION
2654 032440  004737  044230                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032444  000404                             BR      50$             ;GO TO 50$ IF NO ERROR
     032446  000240                             NOP                     ;RETURN HERE IF ERROR
     032450  104000                             EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032452  000137  033176                     JMP     350$            ;GO TO 350$ IF ERROR
2655 032456                             50$:
2656
2657                                    ;VERIFY NO ERRORS DURING FORMAT
2658 032456  004737  057742                     JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     032462  000405                             BR      60$             ;GO TO 60$ IF NO ERROR
     032464  000240                             NOP                     ;RETURN HERE IF ERROR
     032466  104000                             EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     032470  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032472  000137  033176                     JMP     350$            ;GO TO 350$ IF ERROR
2659 032476                             60$:
2660
2661                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
2662 032476  012737  032506  001124             MOV     #70$,$LPERR
2663 032504  000410                             BR      80$                     ;SKIP TO WRITE OPERATION
2664
2665                                    ;;*****************************************************************
2666                                    ;LOOP #2        WRITE,READ
2667
2668 032506                             70$:
2669
2670
2671                                    ;PREPARE DEVICE FOR WRITE OPERATION
2672 032506  004737  040472                     JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     032512  054130                             .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                        ;VERIFY RECALIBRATION
     032514  000404                             BR      80$             ;GO TO 80$ IF NO ERROR
     032516  000240                             NOP                     ;RETURN HERE IF ERROR
     032520  104000                             EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     032522  000137  033176                     JMP     350$            ;GO TO 350$ IF ERROR
2673 032526                             80$:
2674
2675 032526                             120$:
2676
2677                                    ;WRITE DATA TO THE DRIVE
2678 032526  012737  177000  001412             MOV     #-256.*2,RMWCO          ;2 SECTORS (2'S COMP)
2679 032534  012737  106314  001414             MOV     #BUFONE,RMBAO           ;CHANGE ADDRESS
2680 032542  012737  000061  001410             MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2681 032550  012737  071666  001174             MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
2682 032556  012737  000001  001176             MOV     #1,$TMP1                ;RANGE OF PATTERN
2683 032564  004737  043344                     JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
2684 032570  012702  001551                     MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2685 032574  112722  000006                     MOVB    #RMDA,(R2)+
2686 032600  112722  000034                     MOVB    #RMDC,(R2)+
```

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

                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      130$            ;GO TO 130$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                130$:

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

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

                                ;GO READ STATUS FOR WRITE COMMAND
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      140$            ;GO TO 140$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                140$:

                                ;CHECK FOR ERRORS DURING WRITE OPERATION
                                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                                        BR      150$            ;GO TO 150$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                150$:
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      160$            ;GO TO 160$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                160$:

                                170$:
                                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                                        BR      180$            ;GO TO 180$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     350$            ;GO TO 350$ IF ERROR
                                180$:

                                ;CHANGE LOOP ADDRESSES
                                        MOV     #190$,$LPERR
                                        BR      200$                    ;SKIP TO NEXT OPERATION

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

```
                                        ;LOOP #3        READ
                                        190$:
                                        ;PREPARE DEVICE FOR READ OPERATION
032764  004737  040472                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
032770  054130                                  .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF 'SKI'' OR 'PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
032772  000404                                  BR      200$            ;GO TO 200$ IF NO ERROR
032774  000240                                  NOP                     ;RETURN HERE IF ERROR
032776  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
033000  000137  033176                          JMP     350$            ;GO TO 350$ IF ERROR
033004                                  200$:
033004                                  240$:
                                        ;READ DATA FROM DEVICE
033004  012737  000051  001410                  MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
033012  012702  001551                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
033016  112722  000006                          MOVB    #RMDA,(R2)+
033022  112722  000032                          MOVB    #RMOF,(R2)+
033026  112722  000034                          MOVB    #RMDC,(R2)+
033032  112722  000004                          MOVB    #RMBA,(R2)+
033036  112722  000002                          MOVB    #RMWC,(R2)+
033042  112722  000000                          MOVB    #RMCS1,(R2)+
033046  112712  000200                          MOVB    #200,(R2)
033052  004737  044500                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
033056  000404                                  BR      250$            ;GO TO 250$ IF NO ERROR
033060  000240                                  NOP                     ;RETURN HERE IF ERROR
033062  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
033064  000137  033176                          JMP     350$            ;GO TO 350$ IF ERROR
033070                                  250$:
                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
033070  004737  044144                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
                                        ;WAIT FOR COMMAND TO COMPLETE
033074  004737  045042                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
                                        ;GO READ STATUS FOR READ OPERATION
033100  004737  044230                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
033104  000404                                  BR      260$            ;GO TO 260$ IF NO ERROR
033106  000240                                  NOP                     ;RETURN HERE IF ERROR
033110  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
0331?2  000137  033176                          JMP     350$            ;GO TO 350$ IF ERROR
033116                                  260$:
                                        ;CHECK FOR ERRORS DURING READ OPERATION
033116  004737  045226                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
033122  000405                                  BR      270$            ;GO TO 270$ IF NO ERROR
033124  000240                                  NOP                     ;RETURN HERE IF ERROR
```

```
033126  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
033130  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
033132  000137  033176                  JMP     350$            ;GO TO 350$ IF ERROR
2753 033136                     270$:
2754 033136  004737  057742            JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
033142  000405                          BR      280$            ;GO TO 280$ IF NO ERROR
033144  000240                          NOP                     ;RETURN HERE IF ERROR
033146  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
033150  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
033152  000137  033176                  JMP     350$            ;GO TO 350$ IF ERROR
2755 033156                     280$:
2756
2757 033156                     290$:
2758 033156  004737  046060            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
033162  000405                          BR      300$            ;GO TO 300$ IF NO ERROR
033164  000240                          NOP                     ;RETURN HERE IF ERROR
033166  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
033170  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
033172  000137  033176                  JMP     350$            ;GO TO 350$ IF ERROR
2759 033176                     300$:
2760
2761 033176                     350$:
2762
2763                            ;;*****************************************************************
                                ;*TEST 24        WRITE, READ EARLY PEAK SHIFT
                                ;;*****************************************************************
033176                          TST24:
033176  000004                          SCOPE                   ;SCOPE CALL
033200  000240                          NOP                     ;START OF TEST
033202  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
033206  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
033212  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
033216  012737  000024  001226          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2764
2765                            ;;*****************************************************************
2766                            ;LOOP #1        FORMAT,WRITE,READ
2767
2768 033224                     10$:
2769
2770                            ;PREPARE THE DEVICE FOR FORMAT OPERATION
2771 033224  004737  040472            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
033230  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                ;VERIFY RECALIBRATION
033232  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
033234  000240                          NOP                     ;RETURN HERE IF ERROR
033236  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
033240  000137  034204                  JMP     350$            ;GO TO 350$ IF ERROR
2772 033244                     20$:
2773
2774                            ;LOAD PARAMETERS AND GENERATE DATA BUFFER
2775 033244  012737  000000  001444    MOV     #0,RMDC0                ;CYLINDER = 0
```

```
2776 033252  012737  000000  001416          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
2777 033260  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
2778 033266  012737  177376  001412          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
2779 033274  012737  010000  001442          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
2780 033302  012737  000063  001410          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
2781
2782                                  ;VERIFY THAT SECTOR IS NOT BAD
     033310  004737  041416                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     033314  000405                          BR      25$             ;GO TO 25$ IF NO ERROR
     033316  104401  070230                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     033322  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     033324  000137  034204                  JMP     350$            ;GO TO 350$ IF ERROR
2783 033330                           25$:
2784 033330  012737  072142  001174          MOV     #EARLY,$TMPO            ;STARTING ADDRESS OF PATTERN
2785 033336  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
2786 033344  004737  043344                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
2787
2788                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
2789 033350  012702  001551                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
2790 033354  112722  000034                  MOVB    #RMDC,(R2)+
2791 033360  112722  000006                  MOVB    #RMDA,(R2)+
2792 033364  112722  000004                  MOVB    #RMBA,(R2)+
2793 033370  112722  000002                  MOVB    #RMWC,(R2)+
2794 033374  112722  000032                  MOVB    #RMOF,(R2)+
2795 033400  112722  000000                  MOVB    #RMCS1,(R2)+
2796 033404  112712  000200                  MOVB    #200,(R2)               ;TERMINATE TABLE
2797 033410                           30$:
2798 033410  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     033414  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
     033416  000240                          NOP                     ;RETURN HERE IF ERROR
     033420  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     033422  000137  034204                  JMP     350$            ;GO TO 350$ IF ERROR
2799 033426                           40$:
2800
2801                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     033426  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2802
2803                                  ;WAIT FOR COMMAND TO COMPLETE
     033432  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2804
2805                                  ;GO READ STATUS FOR FORMAT OPERATION
2806 033436  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     033442  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
     033444  000240                          NOP                     ;RETURN HERE IF ERROR
     033446  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     033450  000137  034204                  JMP     350$            ;GO TO 350$ IF ERROR
2807 033454                           50$:
2808
2809                                  ;VERIFY NO ERRORS DURING FORMAT
2810 033454  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     033460  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
     033462  000240                          NOP                     ;RETURN HERE IF ERROR
     033464  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     033466  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033470  000137  034204                  JMP     350$            ;GO TO 350$ IF ERROR
2811 033474                           60$:
2812
```

```
2813                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
2814 033474  012737  033504  001124              MOV     #70$,$LPERR
2815 033502  000410                              BR      80$
2816
2817                                    ;;*****************************************************************
2818                                    ;LOOP #2        WRITE,READ
2819
2820 033504                             70$:
2821
2822                                    ;PREPARE DEVICE FOR WRITE OPERATION
2823 033504  004737  040472                      JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     033510  154130                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                         ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                         ;CLEAR CONTROLLER & SELECT DEVICE
                                                                         ;VERIFY CONTROLLER CLEAR OPERATION
                                                                         ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                         ;VERIFY PACK ACKNOWLEDGE
                                                                         ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                         ;VERIFY RECALIBRATION
     033512  000404                              BR      80$             ;GO TO 80$ IF NO ERROR
     033514  000240                              NOP                     ;RETURN HERE IF ERROR
     033516  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     033520  000137  034204                      JMP     350$            ;GO TO 350$ IF ERROR
2824 033524                             80$:
2825
2826                                    ;WRITE DATA TO THE DRIVE
2827 033524                             120$:
2828 033524  012737  177400  001412              MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
2829 033532  012737  106320  001414              MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
2830 033540  012737  000061  001410              MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2831 033546  012702  001551                      MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2832 033552  112722  000006                      MOVB    #RMDA,(R2)+
2833 033556  112722  000034                      MOVB    #RMDC,(R2)+
2834 033562  112722  000032                      MOVB    #RMOF,(R2)+
2835 033566  112722  000004                      MOVB    #RMBA,(R2)+
2836 033572  112722  000002                      MOVB    #RMWC,(R2)+
2837 033576  112722  000000                      MOVB    #RMCS1,(R2)+
2838 033602  112722  000200                      MOVB    #200,(R2)+              ;TERMINATE TABLE
2839
2840 033606  004737  044500                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     033612  000404                              BR      130$            ;GO TO 130$ IF NO ERROR
     033614  000240                              NOP                     ;RETURN HERE IF ERROR
     033616  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     033620  000137  034204                      JMP     350$            ;GO TO 350$ IF ERROR
2841 033624                             130$:
2842
2843                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     033624  004737  044144                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2844
2845                                    ;WAIT FOR COMMAND TO COMPLETE
     033630  004737  045042                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2846
2847                                    ;GO READ STATUS FOR WRITE COMMAND
2848 033634  004737  044230                      JSR     PC,GET          ;GO READ REGISTEP(S) WITH GET SUBROUTINE
     033640  000404                              BR      140$            ;GO TO 140$ IF NO ERROR
     033642  000240                              NOP                     ;RETURN HERE IF ERROR
     033644  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
```

```
      033646  000137  034204                    JMP     350$            ;GO TO 350$ IF ERROR
 2849 033652                              140$:
 2850
 2851                                     ;CHECK FOR ERRORS DURING WRITE OPERATION
 2852 033652  004737  045226                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      033656  000405                            BR      150$            ;GO TO 150$ IF NO ERROR
      033660  000240                            NOP                     ;RETURN HERE IF ERROR
      033662  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      033664  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      033666  000137  034204                    JMP     350$            ;GO TO 350$ IF ERROR
 2853 033672                              150$:
 2854 033672  004737  057742                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      033676  000405                            BR      160$            ;GO TO 160$ IF NO ERROR
      033700  000240                            NOP                     ;RETURN HERE IF ERROR
      033702  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      033704  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      033706  000137  034204                    JMP     350$            ;GO TO 350$ IF ERROR
 2855 033712                              160$:
 2856 033712  004737  046060                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      033716  000405                            BR      180$            ;GO TO 180$ IF NO ERROR
      033720  000240                            NOP                     ;RETURN HERE IF ERROR
      033722  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      033724  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      033726  000137  034204                    JMP     350$            ;GO TO 350$ IF ERROR
 2857 033732                              180$:
 2858
 2859                                     ;CHANGE LOOP ADDRESSES
 2860 033732  012737  033742  001124            MOV     #190$,$LPERR
 2861 033740  000410                            BR      200$
 2862
 2863                                     ;;*****************************************************************
 2864                                     ;LOOP #3        READ
 2865
 2866 033742                              190$:
 2867
 2868                                     ;PREPARE DEVICE FOR READ OPERATION
 2869 033742  004737  040472                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      033746  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF ''SKI'' OR ''PIP'' IS SET
                                                                        ;VERIFY RECALIBRATION
      033750  000404                            BR      200$            ;GO TO 200$ IF NO ERROR
      033752  000240                            NOP                     ;RETURN HERE IF ERROR
      033754  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      033756  000137  034204                    JMP     350$            ;GO TO 350$ IF ERROR
 2870 033762                              200$:
 2871
 2872                                     ;READ DATA FROM DEVICE
 2873 033762                              240$:
 2874 033762  012737  000071  001410            MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
 2875 033770  012737  107324  001414            MOV     #BUFTWO+4,RMBA0         ;CHANGE BUFFER
 2876 033776  012702  001551                    MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2877 034002  112722  000006                    MOVB    #RMDA,(R2)+
```

```
2878 034006  112722  000032                MOVB    #RMOF,(R2)+
2879 034012  112722  000034                MOVB    #RMDC,(R2)+
2880 034016  112722  000004                MOVB    #RMBA,(R2)+
2881 034022  112722  000002                MOVB    #RMWC,(R2)+
2882 034026  112722  000000                MOVB    #RMCS1,(R2)+
2883 034032  112712  000200                MOVB    #200,(R2)
2884
2885 034036  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     034042  000404                        BR      250$            ;GO TO 250$ IF NO ERROR
     034044  000240                        NOP                     ;RETURN HERE IF ERROR
     034046  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     034050  000137  034204                JMP     350$            ;GO TO 350$ IF ERROR
2886 034054                        250$:
2887
2888                               ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     034054  004737  044144                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2889
2890                               ;WAIT FOR COMMAND TO COMPLETE
     034060  004737  045042                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2891
2892                               ;GO READ STATUS FOR READ OPERATION
2893 034064  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     034070  000404                        BR      260$            ;GO TO 260$ IF NO ERROR
     034072  000240                        NOP                     ;RETURN HERE IF ERROR
     034074  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     034076  000137  034204                JMP     350$            ;GO TO 350$ IF ERROR
2894 034102                        260$:
2895
2896                               ;CHECK FOR ERRORS DURING READ OPERATION
2897 034102  004737  045226                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     034106  000405                        BR      270$            ;GO TO 270$ IF NO ERROR
     034110  000240                        NOP                     ;RETURN HERE IF ERROR
     034112  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     034114  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     034116  000137  034204                JMP     350$            ;GO TO 350$ IF ERROR
2898 034122                        270$:
2899 034122  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     034126  000405                        BR      280$            ;GO TO 280$ IF NO ERROR
     034130  000240                        NOP                     ;RETURN HERE IF ERROR
     034132  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     034134  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     034136  000137  034204                JMP     350$            ;GO TO 350$ IF ERROR
2900 034142                        280$:
2901 034142  004737  046060                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     034146  000405                        BR      300$            ;GO TO 300$ IF NO ERROR
     034150  000240                        NOP                     ;RETURN HERE IF ERROR
     034152  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     034154  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     034156  000137  034204                JMP     350$            ;GO TO 350$ IF ERROR
2902 034162                        300$:
2903 034162  004737  043602                JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
     034166  106320                        .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
     034170  107324                        .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
     034172  000404                        BR      310$            ;GO TO 310$ IF NO ERROR
     034174  000240                        NOP                     ;RETURN HERE IF ERROR
     034176  104000                        EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
     034200  000137  034204                JMP     350$            ;GO TO 350$ IF ERROR
```

```
2904 034204                          310$:
2905
2906 034204                          350$:
2907
2908                                 ;;*****************************************************************
                                     ;*TEST 25       WRITE, READ MIXED PEAK SHIFT
                                     ;;*****************************************************************
     034204                          TST25:
     034204  000004                         SCOPE                   ;SCOPE CALL
     034206  000240                         NOP                     ;START OF TEST
     034210  012706  001100                 MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     034214  013700  001276                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     034220  013701  001464                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     034224  012737  000025  001226         MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2909
2910                                 ;;*****************************************************************
2911                                 ;LOOP #1       FORMAT,WRITE,READ
2912
2913 034232                          10$:
2914
2915                                 ;PREPARE THE DEVICE FOR FORMAT OPERATION
2916 034232  004737  040472                 JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     034236  154130                         .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                     ;VERIFY RECALIBRATION
     034240  000404                         BR      20$             ;GO TO 20$ IF NO ERROR
     034242  000240                         NOP                     ;RETURN HERE IF ERROR
     034244  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     034246  000137  035212                 JMP     350$            ;GO TO 350$ IF ERROR
2917 034252                          20$:
2918
2919                                 ;LOAD PARAMETERS AND GENERATE DATA BUFFER
2920 034252  012737  000000  001444         MOV     #0,RMDCO                ;CYLINDER = 0
2921 034260  012737  000000  001416         MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
2922 034266  012737  106314  001414         MOV     #BUFONE,RMBAO           ;BUS ADDRESS
2923 034274  012737  177376  001412         MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
2924 034302  012737  010000  001442         MOV     #FMT16,RMOFO            ;16 BIT FORMAT
2925 034310  012737  000063  001410         MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
2926
2927                                 ;VERIFY THAT SECTOR IS NOT BAD
     034316  004737  041416                 JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     034322  000405                         BR      25$             ;GO TO 25$ IF NO ERROR
     034324  104401  070230                 TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     034330  104000                         EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     034332  000137  035212                 JMP     350$            ;GO TO 350$ IF ERROR
2928 034336                          25$:
2929 034336  012737  071626  001174         MOV     #MIXED,$TMP0            ;STARTING ADDRESS OF PATTERN
2930 034344  012737  000200  001176         MOV     #128.,$TMP1             ;RANGE OF PATTERN
2931 034352  004737  043344                 JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
2932
2933                                 ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
2934 034356  012702  001551                 MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
```

```
 2935 034362  112722  000034                MOVB    #RMDC,(R2)+
 2936 034366  112722  000006                MOVB    #RMDA,(R2)+
 2937 034372  112722  000004                MOVB    #RMBA,(R2)+
 2938 034376  112722  000002                MOVB    #RMWC,(R2)+
 2939 034402  112722  000032                MOVB    #RMOF,(R2)+
 2940 034406  112722  000000                MOVB    #RMCS1,(R2)+
 2941 034412  112712  000200                MOVB    #200,(R2)               ;TERMINATE TABLE
 2942 034416                          30$:
 2943 034416  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      034422  000404                        BR      40$             ;GO TO 40$ IF NO ERROR
      034424  000240                        NOP                     ;RETURN HERE IF ERROR
      034426  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      034430  000137  035212                JMP     350$            ;GO TO 350$ IF ERROR
 2944 034434                          40$:
 2945
 2946                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      034434  004737  044144                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2947
 2948                                 ;WAIT FOR COMMAND TO COMPLETE
      034440  004737  045042                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2949
 2950
 2951 034444  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      034450  000404                        BR      50$             ;GO TO 50$ IF NO ERROR
      034452  000240                        NOP                     ;RETURN HERE IF ERROR
      034454  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      034456  000137  035212                JMP     350$            ;GO TO 350$ IF ERROR
 2952 034462                          50$:
 2953
 2954                                 ;VERIFY NO ERRORS DURING FORMAT
 2955 034462  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      034466  000405                        BR      60$             ;GO TO 60$ IF NO ERROR
      034470  000240                        NOP                     ;RETURN HERE IF ERROR
      034472  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      034474  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      034476  000137  035212                JMP     350$            ;GO TO 350$ IF ERROR
 2956 034502                          60$:
 2957
 2958                                 ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2959 034502  012737  034512  001124        MOV     #70$,$LPERR
 2960 034510  000410                        BR      80$                     ;SKIP TO WRITE OPERATION
 2961
 2962                                 ;;*****************************************************************
 2963                                 ;LOOP #2        WRITE,READ
 2964
 2965 034512                          70$:
 2966
 2967                                 ;PREPARE DEVICE FOR WRITE OPERATION
 2968 034512  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      034516  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
```

```
034520  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
034522  000240                          NOP                     ;RETURN HERE IF ERROR
034524  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
034526  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
2969  034532                    80$:
2970
2971                              ;WRITE DATA TO THE DRIVE
2972  034532
2973  034532  012737  177400  001412  120$:   MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
2974  034540  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
2975  034546  012737  000061  001410          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2976  034554  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2977  034560  112722  000006                  MOVB    #RMDA,(R2)+
2978  034564  112722  000034                  MOVB    #RMDC,(R2)+
2979  034570  112722  000032                  MOVB    #RMOF,(R2)+
2980  034574  112722  000004                  MOVB    #RMBA,(R2)+
2981  034600  112722  000002                  MOVB    #RMWC,(R2)+
2982  034604  112722  000000                  MOVB    #RMCS1,(R2)+
2983  034610  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
2984
2985  034614  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
034620  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
034622  000240                          NOP                     ;RETURN HERE IF ERROR
034624  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
034626  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
2986  034632                    130$:
2987
2988                              ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
034632  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2989
2990                              ;WAIT FOR COMMAND TO COMPLETE
034636  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2991
2992                              ;GO READ STATUS FOR WRITE COMMAND
2993  034642  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WI'H GET SUBROUTINE
034646  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
034650  000240                          NOP                     ;RETURN HERE IF ERROR
034652  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
034654  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
2994  034660                    140$:
2995
2996                              ;CHECK FOR ERRORS DURING WRITE OPERATION
2997  034660  004737  045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
034664  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
034666  000240                          NOP                     ;RETURN HERE IF ERROR
034670  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
034672  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
034674  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
2998  034700                    150$:
2999  034700  004737  057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
034704  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
034706  000240                          NOP                     ;RETURN HERE IF ERROR
034710  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
034712  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
034714  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
3000  034720                    160$:
3001  034720  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
```

```
        034724  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
        034726  000240                          NOP                     ;RETURN HERE IF ERROR
        034730  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        034732  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        034734  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
  3002  034740                          180$:
  3003
  3004                                  ;CHANGE LOOP ADDRESSES
  3005  034740  012737  034750  001124          MOV     #190$,$LPERR
  3006  034746  000410                          BR      200$                    ;SKIP TO NEXT OPERATION
  3007
  3008                                  ;;*****************************************************************
  3009                                  ;LOOP #3        READ
  3010
  3011  034750                          190$:
  3012
  3013                                  ;PREPARE DEVICE FOR READ OPERATION
  3014  034750  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        034754  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF 'SKI' OR 'PIP' IS SET
                                                                        ;VERIFY RECALIBRATION
        034756  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
        034760  000240                          NOP                     ;RETURN HERE IF ERROR
        034762  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        034764  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
  3015  034770                          200$:
  3016
  3017                                  ;READ DATA FROM DEVICE
  3018  034770                          240$:
  3019  034770  012737  107324  001414          MOV     #BUFTWO+4,RMBAO         ;CHANGE BUFFER
  3020  034776  012737  000071  001410          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
  3021  035004  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  3022  035010  112722  000006                  MOVB    #RMDA,(R2)+
  3023  035014  112722  000032                  MOVB    #RMOF,(R2)+
  3024  035020  112722  000034                  MOVB    #RMDC,(R2)+
  3025  035024  112722  000004                  MOVB    #RMBA,(R2)+
  3026  035030  112722  000002                  MOVB    #RMWC,(R2)+
  3027  035034  112722  000000                  MOVB    #RMCS1,(R2)+
  3028  035040  112712  000200                  MOVB    #200,(R2)
  3029
  3030  035044  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        035050  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
        035052  000240                          NOP                     ;RETURN HERE IF ERROR
        035054  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        035056  000137  035212                  JMP     350$            ;GO TO 350$ IF ERROR
  3031  035062                          250$:
  3032
  3033                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        035062  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  3034
  3035                                  ;WAIT FOR COMMAND TO COMPLETE
        035066  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
```

```
3036
3037                                    ;GO READ STATUS FOR READ OPERATION
3038 035072  004737  044230                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     035076  000404                            BR      260$            ;GO TO 260$ IF NO ERROR
     035100  000240                            NOP                     ;RETURN HERE IF ERROR
     035102  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     035104  000137  035212                    JMP     350$            ;GO TO 350$ IF ERROR
3039 035110                             260$:
3040
3041                                    ;CHECK FOR ERRORS DURING READ OPERATION
3042 035110  004737  045226                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     035114  000405                            BR      270$            ;GO TO 270$ IF NO ERROR
     035116  000240                            NOP                     ;RETURN HERE IF ERROR
     035120  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     035122  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     035124  000137  035212                    JMP     350$            ;GO TO 350$ IF ERROR
3043 035130                             270$:
3044 035130  004737  057742                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     035134  000405                            BR      280$            ;GO TO 280$ IF NO ERROR
     035136  000240                            NOP                     ;RETURN HERE IF ERROR
     035140  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     035142  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     035144  000137  035212                    JMP     350$            ;GO TO 350$ IF ERROR
3045 035150                             280$:
3046 035150  004737  046060                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     035154  000405                            BR      300$            ;GO TO 300$ IF NO ERROR
     035156  000240                            NOP                     ;RETURN HERE IF ERROR
     035160  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     035162  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     035164  000137  035212                    JMP     350$            ;GO TO 350$ IF ERROR
3047 035170                             300$:
3048 035170  004737  043602                    JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
     035174  106320                            .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
     035176  107324                            .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
     035200  000404                            BR      310$            ;GO TO 310$ IF NO ERROR
     035202  000240                            NOP                     ;RETURN HERE IF ERROR
     035204  104000                            EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
     035206  000137  035212                    JMP     350$            ;GO TO 350$ IF ERROR
3049 035212                             310$:
3050
3051 035212                             350$:
3052
3053                                    ;;*****************************************************************
                                        ;*TEST 26        WRITE, READ EACH TRACK
                                        ;;*****************************************************************
     035212                             TST26:
     035212  000004                            SCOPE                   ;SCOPE CALL
     035214  000240                            NOP                     ;START OF TEST
     035216  012706  001100                    MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     035222  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     035226  013701  001464                    MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     035232  012737  000026  001226            MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3054
3055                                    ;;*****************************************************************
3056                                    ;LOOP #1        FORMAT,WRITE,READ
3057
3058 035240                             10$:
```

```
                                ;PREPARE THE DEVICE FOR FORMAT OPERATION
035240  004737  040472                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
035244  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
035246  000404                      BR      20$             ;GO TO 20$ IF NO ERROR
035250  000240                      NOP                     ;RETURN HERE IF ERROR
035252  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
035254  000137  036240              JMP     350$            ;GO TO 350$ IF ERROR
035260                          20$:

                                ;LOAD PARAMETERS AND GENERATE DATA BUFFER
035260  012737  000000  001444        MOV     #0,RMDCO                ;CYLINDER = 0
035266  012737  000000  001416        MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
035274                          25$:
035274  012737  106314  001414        MOV     #BUFONE,RMBAO           ;BUS ADDRESS
035302  012737  177376  001412        MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
035310  012737  010000  001442        MOV     #FMT16,RMOFO            ;16 BIT FORMAT
035316  012737  000063  001410        MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND

                                ;VERIFY THAT SECTOR IS NOT BAD
035324  004737  041416                JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
035330  000405                      BR      26$             ;GO TO 26$ IF NO ERROR
035332  104401  070230              TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
035336  104000                      EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
035340  000137  036240              JMP     350$            ;GO TO 350$ IF ERROR
035344                          26$:
035344  012737  071626  001174        MOV     #MIXED,$TMP0            ;STARTING ADDRESS OF PATTERN
035352  012737  000200  001176        MOV     #128.,$TMP1             ;RANGE OF PATTERN
035360  004737  043344                JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
035364  012702  001551                MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
035370  112722  000034                MOVB    #RMDC,(R2)+
035374  112722  000006                MOVB    #RMDA,(R2)+
035400  112722  000004                MOVB    #RMBA,(R2)+
035404  112722  000002                MOVB    #RMWC,(R2)+
035410  112722  000032                MOVB    #RMOF,(R2)+
035414  112722  000000                MOVB    #RMCS1,(R2)+
035420  112712  000200                MOVB    #200,(R2)               ;TERMINATE TABLE
035424                          30$:

                                ;FORMAT THE DRIVE
035424  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
035430  000404                      BR      40$             ;GO TO 40$ IF NO ERROR
035432  000240                      NOP                     ;RETURN HERE IF ERROR
035434  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
035436  000137  036240              JMP     350$            ;GO TO 350$ IF ERROR
035442                          40$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
```

```
        035442  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 3095
 3096                                    ;WAIT FOR COMMAND TO COMPLETE
        035446  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 3097
 3098                                    ;GO READ STATUS FOR FORMAT OPERATION
 3099 035452  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      035456  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
      035460  000240                          NOP                     ;RETURN HERE IF ERROR
      035462  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      035464  000137  036240                  JMP     350$            ;GO TO 350$ IF ERROR
 3100 035470                             50$:
 3101
 3102                                    ;VERIFY NO ERRORS DURING FORMAT
 3103 035470  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      035474  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
      035476  000240                          NOP                     ;RETURN HERE IF ERROR
      035500  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      035502  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      035504  000137  036240                  JMP     350$            ;GO TO 350$ IF ERROR
 3104 035510                             60$:
 3105
 3106                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 3107 035510  012737  035520  001124          MOV     #70$,$LPERR
 3108 035516  000410                          BR      80$                     ;SKIP TO WRITE OPERATION
 3109
 3110                                    ;;*****************************************************************
 3111                                    ;LOOP #2        WRITE,READ
 3112
 3113 035520                             70$:
 3114
 3115                                    ;PREPARE DEVICE FOR WRITE OPERATION
 3116 035520  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      035524  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                      ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                      ;CLEAR CONTROLLER & SELECT DEVICE
                                                                      ;VERIFY CONTROLLER CLEAR OPERATION
                                                                      ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                      ;VERIFY PACK ACKNOWLEDGE
                                                                      ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                      ;VERIFY RECALIBRATION
      035526  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
      035530  000240                          NOP                     ;RETURN HERE IF ERROR
      035532  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      035534  000137  036240                  JMP     350$            ;GO TO 350$ IF ERROR
 3117 035540                             80$:
 3118
 3119                                    ;WRITE DATA TO THE DRIVE
 3120 035540                             120$:
 3121 035540  012737  106320  001414          MOV     #BUFONE+4,RMBAO         ;CHANGE BUS ADDRESS
 3122 035546  012737  177400  001412          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
 3123 035554  012737  000061  001410          MOV     #WI!GO,RMCS10           ;WRITE DATA COMMAND
 3124 035562  012702  001551                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 3125 035566  112722  000006                  MOVB    #RMDA,(R2)+
 3126 035572  112722  000034                  MOVB    #RMDC,(R2)+
 3127 035576  112722  000032                  MOVB    #RMOF,(R2)+
 3128 035602  112722  000004                  MOVB    #RMBA,(R2)+
```

```
 3129 035606  112722  000002                MOVB    #RMWC,(R2)+
 3130 035612  112722  000000                MOVB    #RMCS1,(R2)+
 3131 035616  112722  000200                MOVB    #200,(R2)+              ;TERMINATE TABLE
 3132
 3133 035622  004737  044500                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      035626  000404                        BR      130$            ;GO TO 130$ IF NO ERROR
      035630  000240                        NOP                     ;RETURN HERE IF ERROR
      035632  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      035634  000137  036240                JMP     350$            ;GO TO 350$ IF ERROR
 3134 035640                          130$:
 3135
 3136                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      035640  004737  044144                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 3137
 3138                                 ;WAIT FOR COMMAND TO COMPLETE
      035644  004737  045042                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 3139
 3140
 3141 035650  004737  044230                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      035654  000404                        BR      140$            ;GO TO 140$ IF NO ERROR
      035656  000240                        NOP                     ;RETURN HERE IF ERROR
      035660  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      035662  000137  036240                JMP     350$            ;GO TO 350$ IF ERROR
 3142 035666                          140$:
 3143
 3144                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
 3145 035666  004737  045226                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      035672  000405                        BR      150$            ;GO TO 150$ IF NO ERROR
      035674  000240                        NOP                     ;RETURN HERE IF ERROR
      035676  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      035700  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      035702  000137  036240                JMP     350$            ;GO TO 350$ IF ERROR
 3146 035706                          150$:
 3147 035706  004737  057742                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      035712  000405                        BR      160$            ;GO TO 160$ IF NO ERROR
      035714  000240                        NOP                     ;RETURN HERE IF ERROR
      035716  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      035720  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      035722  000137  036240                JMP     350$            ;GO TO 350$ IF ERROR
 3148 035726                          160$:
 3149 035726  004737  046060                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      035732  000405                        BR      180$            ;GO TO 180$ IF NO ERROR
      035734  000240                        NOP                     ;RETURN HERE IF ERROR
      035736  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      035740  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      035742  000137  036240                JMP     350$            ;GO TO 350$ IF ERROR
 3150 035746                          180$:
 3151
 3152                                 ;CHANGE LOOP ADDRESSES
 3153 035746  012737  035756  001124        MOV     #190$,$LPERR
 3154 035754  000410                        BR      200$                    ;SKIP TO NEXT OPERATION
 3155
 3156                                 ;;*****************************************************************************
 3157                                 ;LOOP #3        READ
 3158
 3159 035756                          190$:
 3160
```

```
                                        ;PREPARE DEVICE FOR READ OPERATION
035756  004737  040472                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
035762  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
035764  000404                              BR      200$            ;GO TO 200$ IF NO ERROR
035766  000240                              NOP                     ;RETURN HERE IF ERROR
035770  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
035772  000137  036240                      JMP     350$            ;GO TO 350$ IF ERROR
035776                                  200$:

                                        ;READ DATA FROM DEVICE
035776                                  240$:
035776  012737  107324  001414              MOV     #BUFTWO+4,RMBAO         ;CHANGE BUS ADDRESS
036004  012737  000071  001410              MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
036012  012702  001551                      MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
036016  112722  000006                      MOVB    #RMDA,(R2)+
036022  112722  000032                      MOVB    #RMOF,(R2)+
036026  112722  000034                      MOVB    #RMDC,(R2)+
036032  112722  000004                      MOVB    #RMBA,(R2)+
036036  112722  000002                      MOVB    #RMWC,(R2)+
036042  112722  000000                      MOVB    #RMCS1,(R2)+
036046  112712  000200                      MOVB    #200,(R2)

036052  004737  044500                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
036056  000404                              BR      250$            ;GO TO 250$ IF NO ERROR
036060  000240                              NOP                     ;RETURN HERE IF ERROR
036062  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
036064  000137  036240                      JMP     350$            ;GO TO 350$ IF ERROR
036070                                  250$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
036070  004737  044144                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
036074  004737  045042                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR READ OPERATION
036100  004737  044230                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
036104  000404                              BR      260$            ;GO TO 260$ IF NO ERROR
036106  000240                              NOP                     ;RETURN HERE IF ERROR
036110  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
036112  000137  036240                      JMP     350$            ;GO TO 350$ IF ERROR
036116                                  260$:

                                        ;CHECK FOR ERRORS DURING READ OPERATION
036116  004737  045226                      JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
036122  000405                              BR      270$            ;GO TO 270$ IF NO ERROR
036124  000240                              NOP                     ;RETURN HERE IF ERROR
036126  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
036130  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
036132  000137  036240                      JMP     350$            ;GO TO 350$ IF ERROR
```

```
 3191 036136                                    270$:
 3192 036136  004737  057742                            JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      036142  000405                                    BR      280$            ;GO TO 280$ IF NO ERROR
      036144  000240                                    NOP                     ;RETURN HERE IF ERROR
      036146  104000                                    EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      036150  004736                                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      036152  000137  036240                            JMP     350$            ;GO TO 350$ IF ERROR
 3193 036156                                    280$:
 3194 036156  004737  046060                            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      036162  000405                                    BR      300$            ;GO TO 300$ IF NO ERROR
      036164  000240                                    NOP                     ;RETURN HERE IF ERROR
      036166  104000                                    EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      036170  004736                                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      036172  000137  036240                            JMP     350$            ;GO TO 350$ IF ERROR
 3195 036176                                    300$:
 3196 036176  004737  043602                            JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
      036202  106320                                    .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
      036204  107324                                    .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
      036206  000404                                    BR      310$            ;GO TO 310$ IF NO ERROR
      036210  000240                                    NOP                     ;RETURN HERE IF ERROR
      036212  104000                                    EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
      036214  000137  036240                            JMP     350$            ;GO TO 350$ IF ERROR
 3197 036220                                    310$:
 3198 036220  105237  001417                            INCB    RMDA0+1                 ;ADVANCE TO NEXT TRACK
 3199 036224  123737  001417  001333                    CMPB    RMDA0+1,LSTRK+1         ;DONE ?
 3200 036232  101002                                    BHI     350$                    ;YES!!
 3201 036234  000137  035274                            JMP     25$                     ;TEST NEXT TRACK
 3202 036240                                    350$:
 3203
 3204                                           ;;*******************************************************************
                                                ;*TEST 27        READ, WRITE CHECK MULTIPLE SECTORS IN OFFSET MODE
                                                ;;*******************************************************************
      036240                                    TST27:
      036240  000004                                    SCOPE                   ;SCOPE CALL
      036242  000240                                    NOP                     ;START OF TEST
      036244  012706  001100                            MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      036250  013700  001276                            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      036254  013701  001464                            MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      036260  012737  000027  001226                    MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 3205
 3206                                           ;;*******************************************************************
 3207                                           ;LOOP #1        FORMAT,READ OFFSET,WRITE CHECK OFFSET IN BOTH DIRECTIONS
 3208 036266                                    10$:
 3209
 3210                                           ;PREPARE DEVICE FOR FORMAT OPERATION
 3211 036266  004737  040472                            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      036272  154130                                    .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
      036274  000404                                    BR      20$             ;GO TO 20$ IF NO ERROR
      036276  000240                                    NOP                     ;RETURN HERE IF ERROR
      036300  104000                                    EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
```

```
036302  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
3212
3213                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
3214 036306                             20$:
3215 036306  012737  000000  001444              MOV     #0,RMDCO                ;CYLINDER = 0
3216 036314  012737  000000  001416              MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
3217 036322  012737  106314  001414              MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS
3218 036330  012737  176774  001412              MOV     #-258.*2,RMWCO          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
3219 036336  012737  010000  001442              MOV     #FMT16,RMOFO            ;16 BIT MODE
3220 036344  012737  000063  001410              MOV     #WH!GO,RMCS1O           ;WRITE HEADER
3221
3222                                    ;VERIFY THAT SECTOR IS NOT BAD
036352  004737  041416                          JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
036356  000405                                  BR      25$             ;GO TO 25$ IF NO ERROR
036360  104401  070230                          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
036364  104000                                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
036366  000137  037422                          JMP     350$            ;GO TO 350$ IF ERROR
3223 036372                             25$:
3224 036372  012737  071730  001174              MOV     #ZEROS,$TMP0            ;DATA PATTERN
3225 036400  012737  000001  001176              MOV     #1,$TMP1
3226 036406  004737  043344                      JSR     PC,GENBUF               ;TO GENERATE BUFFER FOR FORMAT
3227
3228                                    ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
3229 036412  012702  001551                      MOV     #PUTINX,R2              ;BYTE ENTRY TABLE
3230 036416  112722  000034                      MOVB    #RMDC,(R2)+
3231 036422  112722  000006                      MOVB    #RMDA,(R2)+
3232 036426  112722  000004                      MOVB    #RMBA,(R2)+
3233 036432  112722  000002                      MOVB    #RMWC,(R2)+
3234 036436  112722  000032                      MOVB    #RMOF,(R2)+
3235 036442  112722  000000                      MOVB    #RMCS1,(R2)+
3236 036446  112712  000200                      MOVB    #200,(R2)               ;TABLE TERMINATOR
3237 036452                             30$:
3238 036452  004737  044500                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
036456  000404                                  BR      40$             ;GO TO 40$ IF NO ERROR
036460  000240                                  NOP                     ;RETURN HERE IF ERROR
036462  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
036464  000137  037422                          JMP     350$            ;GO TO 350$ IF ERROR
3239 036470                             40$:
3240
3241                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
036470  004737  044144                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
3242
3243                                    ;WAIT FOR COMMAND TO COMPLETE
036474  004737  045042                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
3244
3245                                    ;GO READ STATUS FOR FORMAT OPERATION
3246 036500  004737  044230                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
036504  000404                                  BR      50$             ;GO TO 50$ IF NO ERROR
036506  000240                                  NOP                     ;RETURN HERE IF ERROR
036510  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
036512  000137  037422                          JMP     350$            ;GO TO 350$ IF ERROR
3247 036516                             50$:
3248
3249                                    ;VERIFY NO ERROR DURING FORMAT
3250 036516  004737  057742                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
036522  000405                                  BR      60$             ;GO TO 60$ IF NO ERROR
036524  000240                                  NOP                     ;RETURN HERE IF ERROR
```

```
036526  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
036530  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
036532  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
3251
3252                                    ;MOV LOOP ADDRESSES TO NEXT OPERATION
3253 036536                             60$:
3254 036536  012737  036560  001124          MOV     #70$,$LPERR     ;ERROR LOOP ADDRESS
3255
3256                                    ;LOCATE BUFFER ADDRESS AND WORD COUNT
3257 036544  012737  177000  001412          MOV     #-256.*2,RMWCO  ;TWO SECTORS
3258 036552  012737  106314  001414          MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS
3259                                    ;;*****************************************************************
3260                                    ;LOOP 2  READ AND WRITE CHECK IN OFFSET MODE
3261 036560                             70$:
3262                                    ;PREPARE DEVICE FOR READ OFFSET OPEATION
3263 036560  004737  040472                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
036564  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
036566  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
036570  000240                          NOP                     ;RETURN HERE IF ERROR
036572  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
036574  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
3264 036600                             80$:
3265 036600                             120$:
3266                                    ;READ OFFSET
3267 036600  012702  001551                  MOV     #PUTINX,R2      ;LOAD THE TABLE
3268 036604  112722  000006                  MOVB    #RMDA,(R2)+
3269 036610  112722  000034                  MOVB    #RMDC,(R2)+
3270 036614  112722  000032                  MOVB    #RMOF,(R2)+
3271 036620  112722  000004                  MOVB    #RMBA,(R2)+
3272 036624  112722  000002                  MOVB    #RMWC,(R2)+
3273 036630  112722  000200                  MOVB    #200,(R2)+      ;TABLE TERMINATOR
3274
3275 036634  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
036640  000404                          BR      125$            ;GO TO 125$ IF NO ERROR
036642  000240                          NOP                     ;RETURN HERE IF ERROR
036644  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
036646  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
3276 036652  012737  000015  001410  125$:   MOV     #OFFSET!GO,RMCS10       ;OFFSET COMMAND
3277 036660  012702  001551                  MOV     #PUTINX,R2
3278 036664  112722  000000                  MOVB    #RMCS1,(R2)+
3279 036670  112722  000200                  MOVB    #200,(R2)+
3280
3281 036674  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
036700  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
036702  000240                          NOP                     ;RETURN HERE IF ERROR
036704  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
036706  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
3282 036712                             130$:
3283
3284                                    ;WAIT FOR COMMAND TO COMPLETE
```

```
                036712  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 3285
 3286           036716  012737  000071  001410          MOV     #RD!GO,RMCS1O           ;READ DATA COMMAND
 3287           036724  012702  001551                  MOV     #PUTINX,R2              ;TABLE INDEX
 3288           036730  112722  000000                  MOVB    #RMCS1,(R2)+
 3289           036734  112712  000200                  MOVB    #200,(R2)               ;TABLE TERMINATOR
 3290
 3291           036740  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                036744  000404                          BR      135$            ;GO TO 135$ IF NO ERROR
                036746  000240                          NOP                     ;RETURN HERE IF ERROR
                036750  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                036752  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3292           036756                          135$:
 3293
 3294                                           ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
                036756  004737  044144                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 3295
 3296                                           ;WAIT FOR COMMAND TO COMPLETE
                036762  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 3297
 3298
 3299           036766  004737  044230                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                036772  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
                036774  000240                          NOP                     ;RETURN HERE IF ERROR
                036776  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                037000  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3300
 3301                                           ;CHECK ERROR DURING READ OFFSET
 3302           037004                          140$:
 3303           037004  004737  045226                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                037010  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                037012  000240                          NOP                     ;RETURN HERE IF ERROR
                037014  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                037016  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                037020  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3304           037024                          150$:
 3305           037024  004737  057742                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                037030  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
                037032  000240                          NOP                     ;RETURN HERE IF ERROR
                037034  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                037036  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                037040  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3306           037044                          160$:
 3307           037044  004737  046060                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                037050  000405                          BR      180$            ;GO TO 180$ IF NO ERROR
                037052  000240                          NOP                     ;RETURN HERE IF ERROR
                037054  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                037056  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                037060  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3308           037064                          180$:
 3309
 3310                                           ;CHANGE LOOP ADDRESS
 3311           037064  012737  037074  001124          MOV     #190$,$LPERR
 3312           037072  000410                          BR      200$            ;TO NEXT OPERATION
 3313
 3314                                           ;;*****************************************************************
 3315                                           ;LOOP 3 WRITE CHECK OFFSET
```

```
 3316 037074                          190$:
 3317
 3318
 3319 037074  004737  040472          ;PREPARE DEVICE FOR READ OPERATION
      037100  154130                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                                              .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                      ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                      ;CLEAR CONTROLLER & SELECT DEVICE
                                                                      ;VERIFY CONTROLLER CLEAR OPERATION
                                                                      ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                      ;VERIFY PACK ACKNOWLEDGE
                                                                      ;RECALIBRATE IF ''SKI'' OR 'PIP'' IS SET
                                                                      ;VERIFY RECALIBRATION
      037102  000404                          BR      200$            ;GO TO 200$ IF NO ERROR
      037104  000240                          NOP                     ;RETURN HERE IF ERROR
      037106  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      037110  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3320 037114                          200$:
 3321
 3322                                 ;WRITE CHECK DATA IN OFFSET MODE
 3323 037114                          240$:
 3324 037114  012702  001551                  MOV     #PUTINX,R2
 3325 037120  112722  000006                  MOVB    #RMDA,(R2)+
 3326 037124  112722  000032                  MOVB    #RMOF,(R2)+
 3327 037130  112722  000034                  MOVB    #RMDC,(R2)+
 3328 037134  112722  000004                  MOVB    #RMBA,(R2)+
 3329 037140  112722  000002                  MOVB    #RMWC,(R2)+
 3330 037144  112712  000200                  MOVB    #200,(R2)       ;TABLE TERMINATOR
 3331
 3332 037150  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      037154  000404                          BR      245$            ;GO TO 245$ IF NO ERROR
      037156  000240                          NOP                     ;RETURN HERE IF ERROR
      037160  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      037162  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3333
 3334                                 ;SETUP OFFSET MODE FIRST
 3335 037166                          245$:
 3336 037166  012737  000015  001410          MOV     #OFFSET!GO,RMCS1O       ;OUTPUT BUFFER
 3337 037174  012702  001551                  MOV     #PUTINX,R2
 3338 037200  112722  000000                  MOVB    #RMCS1,(R2)+
 3339 037204  112712  000200                  MOVB    #200,(R2)
 3340
 3341 037210  004737  044500                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      037214  000404                          BR      250$            ;GO TO 250$ IF NO ERROR
      037216  000240                          NOP                     ;RETURN HERE IF ERROR
      037220  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      037222  000137  037422                  JMP     350$            ;GO TO 350$ IF ERROR
 3342 037226                          250$:
 3343
 3344                                 ;WAIT FOR COMMAND TO COMPLETE
      037226  004737  045042                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 3345
 3346                                 ;EXECUTE WRITE CHECK DATA COMMAND
 3347 037232  012737  000051  001410          MOV     #WCD!GO,RMCS1O
 3348 037240  012702  001551                  MOV     #PUTINX,R2
 3349 037244  112722  000000                  MOVB    #RMCS1,(R2)+
 3350 037250  112712  000200                  MOVB    #200,(R2)
 3351
```

```
3352 037254  004737  044500          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     037260  000404                  BR      255$            ;GO TO 255$ IF NO ERROR
     037262  000240                  NOP                     ;RETURN HERE IF ERROR
     037264  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     037266  000137  037422          JMP     350$            ;GO TO 350$ IF ERROR
3353 037272                      255$:
3354
3355                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     037272  004737  044144          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
3356
3357                             ;WAIT FOR COMMAND TO COMPLETE
     037276  004737  045042          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
3358
3359                             ;READ STATUS FOR WRITE CHECK OFFSET OPERATION
3360 037302  004737  044230          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     037306  000404                  BR      260$            ;GO TO 260$ IF NO ERROR
     037310  000240                  NOP                     ;RETURN HERE IF ERROR
     037312  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     037314  000137  037422          JMP     350$            ;GO TO 350$ IF ERROR
3361 037320                      260$:
3362
3363                             ;CHECK ERROR DURING WRITE CHECK OFFSET OPERATION
3364 037320  004737  045226          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     037324  000405                  BR      270$            ;GO TO 270$ IF NO ERROR
     037326  000240                  NOP                     ;RETURN HERE IF ERROR
     037330  104000                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     037332  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     037334  000137  037422          JMP     350$            ;GO TO 350$ IF ERROR
3365 037340                      270$:
3366 037340  004737  057742          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     037344  000405                  BR      280$            ;GO TO 280$ IF NO ERROR
     037346  000240                  NOP                     ;RETURN HERE IF ERROR
     037350  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     037352  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     037354  000137  037422          JMP     350$            ;GO TO 350$ IF ERROR
3367 037360                      280$:
3368 037360  004737  046060          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     037364  000405                  BR      300$            ;GO TO 300$ IF NO ERROR
     037366  000240                  NOP                     ;RETURN HERE IF ERROR
     037370  104000                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     037372  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     037374  000137  037422          JMP     350$            ;GO TO 350$ IF ERROR
3369 037400                      300$:
3370 037400  032737  000200  001442      BIT     #OFD,RMOFO      ;IN FORWARD DIRECTION OF OFFSET ?
3371 037406  001005                      BNE     350$            ;BRANCH IF SO
3372 037410  052737  000200  001442      BIS     #OFD,RMOFO      ;SET TO FORWARD DIRECTION
3373 037416  000137  036536              JMP     60$             ;CHANGE OFFSET DIRECTION TEST AGAIN
3374 037422                      350$:
3375
```

```
                        .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.
                        ;
037422  000004          $EOSP:  SCOPE
037424  000240                  NOP
037426  013700  001464          MOV     TSTQUE,R0       ;GET POINTER TO TSTQUE
037432  062700  000002          ADD     #2,R0           ;ADJUST POINTER TO NEXT DEVICE
037436  010037  001464          MOV     R0,TSTQUE       ;SAVE POINTER TO TSTQUE
037442  005710                  TST     (R0)            ;ANY MORE DEVICES FOR TEST ?
037444  001402                  BEQ     1$              ;BR IF NO
037446  000137  064076          JMP     SHUT            ;JUMP TO SHUT AND CHECK FOR CONTROL C
037452  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
                        ;
037460                  $EOP:
037460  000240                  NOP
037462  005037  001116          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
037466  005037  001206          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
037472  005237  001230          INC     $PASS           ;;INCREMENT THE PASS NUMBER
037476  042737  100000  001230  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
037504  005327                  DEC     (PC)+           ;;LOOP?
037506  000001          $EOPCT: .WORD   1
037510  003063                  BGT     $DOAGN          ;;YES
037512  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
037514  000001          $ENDCT: .WORD   1
037516  037506                  $EOPCT
037520  104401  037526          TYPE    ,65$            ;;TYPE ASCIZ STRING
037524  000407                  BR      64$             ;;GET OVER THE ASCIZ
                        ;;65$:  .ASCIZ  <12><15>/END PASS #/
037544                  64$:
037544  013746  001230          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                        ;;TYPE PASS NUMBER
037550  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037552  104401  037560          TYPE    ,67$            ;;TYPE ASCIZ STRING
037556  000421                  BR      66$             ;;GET OVER THE ASCIZ
                        ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
037622                  66$:
037622  013746  001126          MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                        ;;TOTAL NUMBER OF ERRORS
037626  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037630  104401  001217          TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
037634  005037  001126          CLR     $ERTTL          ;;CLEAR ERROR TOTAL
037640  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
```

```
037644  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
037646  000005                          RESET                   ;;CLEAR THE WORLD
037650  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
037652  000240                          NOP                     ;;SAVE ROOM
037654  000240                          NOP                     ;;FOR
037656  000240                          NOP                     ;;ACT11
037660                          $DOAGN:
037660  000137                          JMP     @(PC)+          ;;RETURN
037662  064076                  $RTNAD: .WORD   SHUT
037664     377      377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        .EVEN
```

21

```
                                        .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 ASSEMBLED, FORMATTED AND PRINTED ON
                                        ;*ONE OR MORE SUCCEEDING LINES;
                                        ;*      .PAIRED LINES OF ERROR HEADERS AND ERROR DATA
                                        ;*ARE PRINTED AFTER THE ERROR MESSAGE.

037670                                  ERRTYP:
037670  104414                                  SAVREG
037672  032777  020000  141254                  BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
037700  001402                                  BEQ     1$              ;NO!!
037702  000137  040420                          JMP     21$             ;YES!!
                                        ;TYPE UNIT NUMBER, TEST NUMBER, ERROR NUMBER, AND PROGRAM COUNTER
037706  104401  001217                  1$:     TYPE    ,$CRLF
037712  104401  040434                          TYPE    ,ERTY00         ;TYPE 'UNT#''
037716  013746  001234                          MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                        ;;TYPE UNIT NUMBER
037722  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
037724     003                                  .BYTE   3               ;;TYPE 3 DIGIT(S)
037725     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
037726  005037  040424                          CLR     TSTNMB          ;LOAD TEST NUMBER FOR
037732  013737  001226  040424                  MOV     $TESTN,TSTNMB
037740  104401  040442                          TYPE    ,ERTY01         ;TYPE 'TST#''
037744  013746  040424                          MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                        ;;TYPE TEST NUMBER
037750  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
037752     003                                  .BYTE   3               ;;TYPE 3 DIGIT(S)
037753     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
037754  005037  040426                          CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
037760  113737  001130  040426                  MOVB    $ITEMB,ERRNMB   ;TYPEOUT
037766  001406                                  BEQ     2$              ;SKIP IF NO ERROR CALLED
037770  104401  040452                          TYPE    ,ERTY02         ;TYPE 'ERR#''
037774  013746  040426                          MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                        ;;TYPE ERROR NUMBER
040000  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
040002     003                                  .BYTE   3               ;;TYPE 3 DIGIT(S)
040003     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
040004  104401  040461          2$:     TYPE    ,ERTY03         ;TYPE 'PC='
040010  013746  001132                          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;TYPE PROGRAM COUNTER
040014  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
040016     006                                  .BYTF   6               ;;TYPE 6 DIGIT(S)
040017     001                                  .BYTE   1               ;;TYPE LEADING ZEROS
                                        ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
040020  005737  040426                  3$:     TST     ERRNMB          ;WAS AN ERROR CALLED?
040024  001575                                  BEQ     21$             ;NO!!
040026  104401  001217                          TYPE    ,$CRLF          ;YES-TYPE CRLF
040032  105037  040432                          CLRB    BOTFLG          ;CLEAR BOT FLAG
040036  105037  040433                          CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
040042  013700  040426                          MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
```

```
040046  006300                          ASL     R0              ;FOUR ENTRIES IN ERROR
040050  006300                          ASL     R0              ;TABLE
040052  006300                          ASL     R0
040054  062700  001570                  ADD     #$ERRTB-8.,R0
040060  011001                          MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
                                                                ;TABLE
040062  001507                          BEQ     13$             ;BRANCH IF NO ERROR MESSAGE
                                ;TYPE THE ERROR MESSAGE
040064  012102                  4$:     MOV     (R1)+,R2        ;R2=ADDRESS OF MESSAGE STRING
040066  001505                          BEQ     12$             ;BRANCH IF END OF MESSAGE
040070  010237  040236                  MOV     R2,11$          ;LOAD ADDRESS OF STRING
040074  005037  040430                  CLR     BOTADR          ;CLEAR BOT ADDRESS
040100  112203                  5$:     MOVB    (R2)+,R3        ;END OF STRING??
040102  001454                          BEQ     10$             ;YES!!
040104  122703  000015                  CMPB    #CR,R3          ;CARRIAGE RETURN??
040110  001003                          BNE     6$              ;NO!!
040112  105037  040433                  CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
040116  000770                          BR      5$              ;GET NEXT CHARACTER
040120  122703  000012          6$:     CMPB    #LF,R3          ;LINE FEED??
040124  001765                          BEQ     5$              ;YES-GET NEXT CHARACTER
040126  122703  000011                  CMPB    #HT,R3          ;HORIZONTAL TAB??
040132  001007                          BNE     8$              ;NO!!
040134  105237  040433          7$:     INCB    CHRCNT          ;ADJUST CHARACTER COUNT
040140  132737  000007  040433          BITB    #7,CHRCNT
040146  001372                          BNE     7$
040150  000407                          BR      9$
040152  105237  040433          8$:     INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
040156  122703  000040                  CMPB    #' ,R3          ;SPACE??
040162  001002                          BNE     9$              ;NO!!
040164  010237  040430                  MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
040170  122737  000100  040433  9$:     CMPB    #64.,CHRCNT     ;END OF LINE??
040176  103340                          BHIS    5$              ;NO!!
040200  013704  040430                  MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
040204  001007                          BNE     90$             ;BRANCH IF SPACE DETECTED
040206  104401  001217                  TYPE    ,$CRLF          ;TYPE CRLF
040212  105037  040433                  CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
040216  013702  040236                  MOV     11$,R2          ;SET UP R2 FOR TESTING
040222  000726                          BR      5$
040224  105044                  90$:    CLRB    -(R4)           ;REPLACE SPACE
040226  112737  177777  040432          MOVB    #-1,BOTFLG      ;SET BOT FLAG
040234  104401                  10$:    TYPE                    ;TYPE ERROR MESSAGE STRING
040236  000000                  11$:    .WORD                   ;STRING ADDRESS GOES HERE
040240  105737  040432                  TSTB    BOTFLG          ;WAS STRING TRUNCATED??
040244  001707                          BEQ     4$              ;NO!!
040246  104401  001217                  TYPE    ,$CRLF          ;YES-TYPE CRLF
040252  105037  040432                  CLRB    BOTFLG          ;CLEAR BOT FLAG
040256  105037  040433                  CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
040262  013702  040430                  MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
040266  010237  040236                  MOV     R2,11$          ;SETUP 11$ FOR TYPING
040272  112742  000040                  MOVB    #' ,-(R2)       ;RESTORE SPACE
040276  105722                          TSTB    (R2)+           ;RESTORE R2
040300  000677                          BR      5$              ;TYPE REST OF STRING
040302                          12$:
                                ;TYPE ERROR HEADER AND ERROR DATA
040302                          13$:
040302  016001  000002                  MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
040306  001444                          BEQ     21$             ;BRANCH IF NO HEADER
```

```
 040310  104401  001217            TYPE    ,$CRLF          ;(ASSUME NO DATA)
 040314  016002  000004            MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
 040320  016003  000006            MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
 040324  012137  040334    14$:    MOV     (R1)+,15$       ;PUT HEADER ADDRESS FOR TYPE
 040330  001433                    BEQ     21$             ;BRANCH IF END OF HEADERS
                                                         ;(ASSUME END OF DATA)
 040332  104401                    TYPE
 040334  000000            15$:    .WORD   0               ;HEADER ADDRESS GOES HERE
 040336  104401  001217            TYPE    ,$CRLF
 040342  005702                    TST     R2              ;DATA WITH HEADER??
 040344  001767                    BEQ     14$             ;NO!!
 040346  012204                    MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
 040350  012305                    MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
 040352  105725            16$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
 040354  100407                    BMI     18$             ;BINARY
 040356  001403                    BEQ     17$             ;OCTAL
 040360  013446                    MOV     @(R4)+,-(SP)    ;DECIMAL
 040362  104405                    TYPDS
 040364  000405                    BR      19$
 040366  013446            17$:    MOV     @(R4)+,-(SP)
 040370  104402                    TYPOC
 040372  000402                    BR      19$
 040374  013446            18$:    MOV     @(R4)+,-(SP)
 040376  104406                    TYPBN
 040400  005714            19$:    TST     (R4)            ;MORE DATA??
 040402  001403                    BEQ     20$             ;NO!!
 040404  104401  040467            TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
 040410  000760                    BR      16$             ;AND CONTINUE
 040412  104401  001217    20$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
 040416  000742                    BR      14$             ;BEFORE NEXT HEADER
 040420  104415            21$:    RESREG
 040422  000207                    RTS     PC

 040424  000000            TSTNMB: .WORD   0               ;TEST NUMBER
 040426  000000            ERRNMB: .WORD   0               ;ERROR NUMBER
 040430  000000            BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
 040432     000            BOTFLG: .BYTE   0               ;BOT FLAG
 040433     000            CHRCNT: .BYTE   0               ;CHARACTER COUNT

 040434     125     116     111    ERTY00: .ASCIZ  @UNIT#@
 040442     054     040     124    ERTY01: .ASCIZ  @, TEST#@
 040452     054     040     105    ERTY02: .ASCIZ  @, ERR#@
 040461     054     040     120    ERTY03: .ASCIZ  @, PC=@
 040467     040     040     000    ERTY04: .ASCIZ  @ @
                                    .EVEN
```

```
                        .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

040472                  TSTPRP:

                        ;STORE TASK DESCRIPTOR AND CLEAR USER'S ERROR CALL
040472  017637  000000  041412          MOV     @(SP),500$      ;STORE DESCRIPTOR
040500  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
040504  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR CALL
040510  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN

040514  004737  044144                  JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
040520  004737  044230                  JSR     PC,GET          ;GET RMER2
040524  000411                        BR      15$             ;BR IF NO ERROR DETECTED
040526  000401                        BR      10$             ;GET OVER ERROR NUMBER
040530  000000                        .WORD   0               ;ERROR DEFINED BY GET SUBROUTINE
040532  062716  000004          10$:    ADD     #4,(SP)         ;XFER ERROR TO USER AND
040536  113776  040530  000000        MOVB    10$-2,@(SP)     ;GET ERROR NUMBER.
040544  000137  041402                JMP     400$

040550  013737  001376  041414  15$:    MOV     RMER2I,505$     ;GET RMER2 AND SAVE FOR LATER

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

```
                                        ;SELECT DEVICE AND VERIFY DEVICE AVAILABLE IF BIT 15 SET IN TASK
040556  005737  041412                          TST     500$            ;SELECT DEVICE??
040562  100014                                  BPL     30$             ;NO!!

040564  004737  052132                          JSR     PC,DEVSEL       ;GO SELECT DEVICE
040570  000411                                  BR      30$             ;NO ERROR - CONTINUE
040572  000401                                  BR      20$
040574  000000                                  .WORD   0               ;ERROR NUMBER FROM DEVSEL
040576  062716  000004              20$:        ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040602  113776  040574  000000                  MOVB    20$-2,@(SP)
040610  000137  041402                          JMP     400$

                                        ;;**************************************************************************
                                        ;CLEAR CONTROLLER IF BIT 14 IS SET IN TASK
040614                                  30$:
040614  032737  040000  041412                  BIT     #BIT14,500$     ;CLEAR CONTROLLER??
040622  001451                                  BEQ     120$            ;NO!!

040624  004737  053604                          JSR     PC,CNTCLR       ;GO CLEAR CONTROLLER
040630  000411                                  BR      60$             ;CONTINUE - NO ERROR
040632  000401                                  BR      50$
040634  000000                          40$:    .WORD   0               ;ERROR NUMBER FROM CNTCLR
040636  062716  000004                  50$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040642  113776  040634  000000                  MOVB    40$,@(SP)
040650  000137  041402                          JMP     400$

                                        ;;**************************************************************************
                                        ;VERIFY CONTROLLER CLEAR IF BIT6 SET IN TASK
040654                                  60$:
040654  032737  000100  041412                  BIT     #BIT6,500$      ;VERIFY??
040662  001431                                  BEQ     120$            ;NO!!

040664  004737  044230                          JSR     PC,GET          ;GO GET STATUS
040670  000411                                  BR      90$             ;NO ERROR GETTING STATUS
040672  000401                                  BR      80$
040674  000000                          70$:    .WORD   0               ;ERROR FROM GETTING STATUS
040676  062716  000004                  80$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
040702  113776  040674  000000                  MOVB    70$,@(SP)
040710  000137  041402                          JMP     400$

040714  004737  053722                  90$:    JSR     PC,CLRSTS       ;GO VERIFY STATUS CLEAR
040720  000412                                  BR      120$            ;NO ERROR IN CLEAR
040722  000401                                  BR      110$
040724  000000                          100$:   .WORD   0               ;ERROR IN STATUS CLEAR
040726  005726                          110$:   TST     (SP)+           ;STRIP RETURN ADDRESS TO
040730  062716  000004                          ADD     #4,(SP)         ;SUBROUTINE AND TRANSFER
040734  113776  040724  000000                  MOVB    100$,@(SP)      ;ERROR TO USER
040742  000137  041402                          JMP     400$

                                        ;;**************************************************************************
                                        ;EXECUTE PACK ACKNOWLEDGE IF BIT12 SET IN TASK AND VOLUME IS
                                        ;NOT VALID
040746                                  120$:
040746  032737  010000  041412                  BIT     #BIT12,500$     ;PACK ACKNOWLEDGE??
040754  001503                                  BEQ     240$            ;NO!!

040756  004737  044230                          JSR     PC,GET
```

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

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

                                        CLR     MEDENB          ;CLEAR MEDIA ENABLE
                                        MOV     #PAKACK!GO,RMCS1O       ;LOAD PACK ACK COMMAND
                                        MOVB    #RMCS1,PUTINX   ;SETUP REGISTER INDEX TABLE
                                        MOVB    #200,PUTINX+1
                                        JSR     PC,PUT          ;GO WRITE COMMAND
                                        BR      180$            ;NO ERROR LOADING REGISTER
                                        BR      170$
                                160$:   .WORD   0               ;ERROR FROM PUT SUB
                                170$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                                        MOVB    160$,@(SP)
                                        BR      400$

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

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

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

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

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

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

```
041206  062716  000004          260$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
041212  113776  041204  000000          MOVB    250$,@(SP)
041220  000470                          BR      400$
041222  032737  040000  041414  270$:   BIT     #SKI,505$       ;WAS SKI SET ?
041230  001004                          BNE     280$            ;YES, GO RECALIBRATE
041232  032737  020000  001346          BIT     #PIP,RMDSI      ;IS PIP ACTIVE??
041240  001462                          BEQ     410$            ;NO!!
041242  012737  000007  001410  280$:   MOV     #RECAL!GO,RMCS10        ;LOAD RECALIBRATE COMMAND
041250  112737  000000  001551          MOVB    #RMCS1,PUTINX   ;AND REGISTER INDEX
041256  112737  000200  001552          MOVB    #200,PUTINX+1
041264  004737  044500                  JSR     PC,PUT          ;GO ISSUE RECALIBRATE
041270  000410                          BR      300$            ;NO ERROR
041272  000401                          BR      290$
041274  000000                          .WORD   0               ;ERROR IN REGISTER TRANSFER
041276  062716  000004          290$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
041302  113776  041274  000000          MOVB    290$-2,@(SP)
041310  000434                          BR      400$
041312  004737  045042          300$:   JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
                                ;;*************************************************************
                                ;VERIFY RECALIBRATE IF BIT 3 SET IN TASK
041316  032737  000010  041412          BIT     #BIT3,500$      ;VERIFY RECALIBRATE??
041324  001430                          BEQ     410$            ;NO!!
041326  004737  044230                  JSR     PC,GET          ;GO GET STATUS
041332  000410                          BR      330$            ;NO ERROR GETTING STATUS
041334  000401                          BR      320$
041336  000000                  310$:   .WORD   0               ;ERROR FROM GET
041340  062716  000004          320$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
041344  113776  041336  000000          MOVB    310$,@(SP)
041352  000413                          BR      400$
041354  004737  055376          330$:   JSR     PC,RCLSTS       ;GO CHECK RECALIBRATE
041360  000412                          BR      410$            ;NO ERROR DURING RECALIBRATE
041362  000401                          BR      350$
041364  000000                  340$:   .WORD   0               ;ERROR DURING RECALIBRATE
041366  005726                  350$:   TST     (SP)+           ;STRIP RETURN TO SUB AND
041370  062716  000004                  ADD     #4,(SP)         ;TRANSFER ERROR TO USER
041374  113776  041364  000000          MOVB    340$,@(SP)
041402  162716  000002          400$:   SUB     #2,(SP)         ;MOVE SP BACK BEFORE ERROR
041406  000240                  410$:   NOP
041410  000207                          RTS     PC              ;RETURN TO USER
041412  000000                  500$:   .WORD   0               ;TASK/VERIFY DESCRIPTOR
041414  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.
                                ;
041416                          BADSCT:
041416  062716  000006                  ADD     #6,(SP)         ;CLEAR ERROR NUMBER IN USER'S
041422  105076  000000                  CLRB    @(SP)           ;ERROR CALL.
041426  162716  000006                  SUB     #6,(SP)
                                ;
                                ;TEST ''MEDIA ENABLE'' TO DETERMINE WHETHER OR NOT THE BAD SECTOR FILES
                                ;HAVE BEEN RECOVERED.
041432  005737  001510                  TST     MEDENB          ;HAS BAD SECTOR FILES BEEN RECOVERED ?
041436  001402                          BEQ     5$              ;BR IF NO
041440  000137  042712                  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
041444                          5$:
041444  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
041446  005000                          CLR     R0              ;START WITH RMCS1
041450  016060  001410  106314  10$:    MOV     PUTBUF(R0),BUFFER(R0)
041456  062700  000002                  ADD     #2,R0           ;ADVANCE TO NEXT BUFFER POSITION
041462  022700  000046                  CMP     #46,R0          ;END OF BUFFER
041466  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)
041470  012737  000003  043330          MOV     #3,500$         ;RETRY COUNT
```

```
     57 041476  012737  001466  001444          MOV     #822.,RMDCO     ;DESIRED CYLINDER = 822.
     58 041504  013737  001332  001416          MOV     LSTRK,RMDAO     ;STARTING LAST TRACK, SECTOR = 0
     59 041512  012737  177376  001412          MOV     #-258.,RMWCO    ;2 + 256. WORDS (2'S COMP)
     60 041520  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
     61 041526  012737  110324  001414          MOV     #MFGFIL,RMBAO   ;POINT TO MANUFACTURES FILE BUFFER
     62
     63 041534  012700  001551                  MOV     #PUTINX,R0      ;R0 POINTS TO REGISTER INDEX TABLE
     64 041540  112720  000006                  MOVB    #RMDA,(R0)+
     65 041544  112720  000034                  MOVB    #RMDC,(R0)+
     66 041550  112720  000002                  MOVB    #RMWC,(R0)+
     67 041554  112720  000032                  MOVB    #RMOF,(R0)+
     68 041560  112720  000004                  MOVB    #RMBA,(R0)+
     69 041564  112720  000000                  MOVB    #RMCS1,(R0)+
     70 041570  112720  000200                  MOVB    #200,(R0)+
     71 041574  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     72
     73                                 ;SET GET INDEX TABLE FOR READING STATUS
     74 041576                          20$:
     75 041576  004737  044144                  JSR     PC,GETSTS       ;SETUP GET INDEX REGISTER FOR STATUS
     76 041602  004737  044230                  JSR     PC,GET          ;GET REGISTERS
     77 041606  000411                          BR      30$             ;BR IF NO ERROR
     78 041610  000401                          BR      25$             ;JUMP OVER ERROR NUMBER
     79 041612  000000                          .WORD   0               ;ERROR DEFINED BY GET SUB
     80 041614  062716  000006          25$:    ADD     #6,(SP)         ;XFER ERROR TO USER AND
     81 041620  113776  041612  000000        MOVB    25$-2,@(SP)     ;GET ERROR NUMBER.
     82 041626  000137  042412                  JMP     215$
     83
     84 041632  013737  001376  043342  30$:    MOV     RMER2I,550$     ;GET RMER2 AND SAVE FOR LATER
     85
     86                                 ;CLEAR THE DEVICE USING DRIVE CLEAR COMMAND
     87 041640  012737  000011  001410          MOV     #DRVCLR!GO,RMCS10       ;LOAD COMMAND IN PUT BUFFER
     88 041646  004737  044500                  JSR     PC,PUT          ;OUTPUT COMMAND
     89 041652  000411                          BR      45$             ;RETURN HERE IF NO ERROR
     90 041654  000401                          BR      40$             ;GET AROUND ERROR #
     91 041656  000000                  35$:    .WORD   0               ;ERROR # GOES HERE
     92 041660  062716  000006          40$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
     93 041664  113776  041656  000000        MOVB    35$,@(SP)       ;MOVE ERROR NUMBER TO USER
     94 041672  000137  042412                  JMP     215$
     95
     96 041676  004737  045042          45$:    JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
     97 041702  004737  044230                  JSR     PC,GET          ;GO GET STATUS
     98 041706  000411                          BR      60$             ;RETURN HERE IF NO ERROR
     99 041710  000401                          BR      55$             ;GET AROUND ERROR #
    100 041712  000000                  50$:    .WORD   0               ;ERROR # GOES HERE
    101 041714  062716  000006          55$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
    102 041720  113776  041712  000000        MOVB    50$,@(SP)       ;MOVE ERROR # TO USERS ERROR CALL
    103 041726  000137  042412                  JMP     215$
    104
    105 041732  004737  057140          60$:    JSR     PC,DRVSTS       ;GO VERIFY DRIVE CLEAR COMMAND
    106 041736  000412                          BR      75$             ;RETURN HERE IF NO ERROR
    107 041740  000401                          BR      70$             ;GET AROUND ERROR
    108 041742  000000                  65$:    .WORD   0               ;ERROR # GOES HERE
    109 041744  005726                  70$:    TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
    110 041746  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
    111 041752  113776  041742  000000        MOVB    65$,@(SP)       ;MOVE ERROR # TO USER CALL
    112 041760  000137  042412                  JMP     215$
    113
```

```
                                        ;ISSUE A PACK ACKNOWLEDGE IF VOLUME VALID IS RESET
                                        75$:
041764 032737 000100 001346             BIT     #VV,RMDSI       ;IS VV RESET ??
041772 001052                           BNE     120$            ;NO !!

041774 012737 000023 001410             MOV     #PACACK!GO,RMCS1O       ;LOAD COMMAND
042002 004737 044500                    JSR     PC,PUT          ;GO PUT COMMAND TO DRIVE
042006 000411                           BR      90$             ;RETURN HERE IF NO OUTPUT ERROR
042010 000401                           BR      85$             ;GET AROUND ERROR #
042012 000000                   80$:    .WORD   0               ;ERROR # GOES HERE
042014 062716 000006            85$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
042020 113776 042012 000000             MOVB    80$,@(SP)       ;MOVE ERROR # TO ERROR CALL
042026 000137 042412                    JMP     215$

042032 004737 045042            90$:    JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
042036 004737 044230                    JSR     PC,GET          ;GO GET STATUS FOR PACK ACK
042042 000411                           BR      105$            ;RETURN HERE IF NO ERROR
042044 000401                           BR      100$            ;GET AROUND ERROR #
042046 000000                   95$:    .WORD   0               ;ERROR # GOES HERE
042050 062716 000006            100$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
042054 113776 042046 000000             MOVB    95$,@(SP)       ;MOVE ERROR # TO CALL
042062 000137 042412                    JMP     215$

042066 004737 054602            105$:   JSR     PC,ACKSTS       ;GO VERIFY ACKNOWLEDGE STATUS
042072 000412                           BR      120$            ;RETURN HERE IF NO ERROR
042074 000401                           BR      115$            ;GET AROUND ERROR #
042076 000000                   110$:   .WORD   0               ;ERROR # GOES HERE
042100 005726                   115$:   TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
042102 062716 000006                    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
042106 113776 042076 000000             MOVB    110$,@(SP)      ;MOVE ERROR # TO USERS ERROR CALL
042114 000137 042412                    JMP     215$

                                        ;RECALIBRATE THE DRIVE IF ''SKI'' OR 'PIP IS SET
042120                                  120$:
042120 032737 040000 043342             BIT     #SKI,550$       ;WAS SKI SET ?
042126 001004                           BNE     125$            ;YES, GO RECALIBRATE
042130 032737 020000 001346             BIT     #PIP,RMDSI      ;IS PIP SET ??
042136 001452                           BEQ     165$            ;NO !!

042140 012737 000007 001410     125$:   MOV     #RECAL!GO,RMCS1O        ;LOAD RECALIBRATE COMMAND
042146 004737 044500                    JSR     PC,PUT          ;PUT THE RECAL COMMAND
042152 000411                           BR      135$            ;RETURN HERE IF NO ERROR
042154 000401                           BR      130$            ;GET AROUND ERROR #
042156 000000                           .WORD   0               ;ERROR # GOES HERE
042160 062716 000006            130$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
042164 113776 042156 000000             MOVB    130$-2,@(SP)    ;MOVE ERROR # TO USERS CALL
042172 000137 042412                    JMP     215$

042176 004737 045042            135$:   JSR     PC,TIMOUT       ;WAIT FOR RECALIBRATE TO COMPLETE
042202 004737 044230                    JSR     PC,GET          ;GO GET RECAL STATUS
042206 000411                           BR      150$            ;RETURN HERE IF NO ERROR
042210 000401                           BR      145$            ;GET AROUND ERROR #
042212 000000                   140$:   .WORD   0               ;ERROR # GOES HERE
042214 062716 000006            145$:   ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
042220 113776 042212 000000             MOVB    140$,@(SP)      ;MOVE ERROR TO USERS CALL
042226 000137 042412                    JMP     215$
```

```
                                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 RMDA0, INCLUDING HEADER AND DATA
                                165$:
                                        MOV     #RH!GO,RMCS10   ;LOAD READ HEADER AND DATA COMMAND
                                        JSR     PC,PUT          ;PUT COMMAND
                                        BR      180$            ;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$,@(SP)      ;MOVE ERRGR # TO USERS ERROR CALL
                                        JMP     215$

                                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
042444  062737  000002  001416          ADD     #2,RMDA0        ;ADVANCE SECTOR ADDRESS BY 2
042452  122737  000012  001416          CMPB    #10.,RMDA0      ;QUIT IF ALL MFG SECTORS HAVE BEEN
042460  001413                          BEQ     230$            ;TRIED.
042462  122737  000040  001416          CMPB    #32.,RMDA0      ;QUIT IF ALL USER SECTORS HAVE BEEN
042470  001407                          BEQ     230$            ;TRIED.

042472  012737  000003  043330          MOV     #3,500$         ;REINSTATE RETRY COUNT FOR THIS SECTOR
042500  162716  000006          225$:   SUB     #6,(SP)         ;MOVE SP BACK TO NO ERROR
042504  000137  041576                  JMP     20$             ;RETRY THE READ OPERATION

                                        ;THE BAD SECTOR FILE CANNOT BE READ
042510                          230$:
042510  000240                          NOP
042512  032777  020000  136434          BIT     #SW13,@SWR      ;INHIBIT MESSAGE ?
042520  001002                          BNE     235$            ;YES
042522  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO ERROR RETURN
042526  000137  043324          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.
042532                          240$:
042532  022737  111332  001414          CMP     #USRFIL,RMBA0   ;WAS THE USER FILE READ ??
042540  001446                          BEQ     260$            ;YES - READ IS COMPLETE
042542  112737  000012  001416          MOVB    #10.,RMDA0      ;READ THE USER FILE LAST TRACK, SECTOR = 10.
042550  012737  111332  001414          MOV     #USRFIL,RMBA0   ;POINT TO USERS FILE BUFFER

042556  012737  000003  043330          MOV     #3,500$         ;RELOAD THE RETRY COUNT FOR THIS SECTOR
042564  000137  041576                  JMP     20$             ;GO READ THE USER FILE

                                        ;DUMMY THE BAD SECTOR FILES
042570                          250$:
042570  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042572  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042574  012701  000374                  MOV     #252.,R1        ;R1 = NUMBER OF ENTRIES IN FILES
042600  012700  000014                  MOV     #14,R0          ;R0 = ADDRESS INDEX TO FILE  STORAGE
042604  012760  177777  110324  255$:   MOV     #-1,MFGFIL(R0)  ;ENTER ALL ONES IN MFG FILE
042612  012760  177777  111332          MOV     #-1,USRFIL(R0)  ;ENTER ALL ONES IN USER FILE
042620  005720                          TST     (R0)+           ;ADVANCE ADDRESS
042622  005301                          DEC     R1              ;DECREMENT COUNT
042624  001367                          BNE     255$            ;CONTINUE IF NOT DONE

042626  012701  000006                  MOV     #6.,R1          ;CLEAR HEADER, CLEAR ID & SERIAL NUMBERS
042632  005000                          CLR     R0
042634  005060  110324          257$:   CLR     MFGFIL(R0)
042640  005060  111332                  CLR     USRFIL(R0)
042644  005720                          TST     (R0)+           ;ADVANCE ADDRESS
042646  005301                          DEC     R1
042650  001371                          BNE     257$
042652  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
042654  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0

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

```
042656                          260$:
042656  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042660  005000                          CLR     R0              ;R0 IS REGISTER INDEX
042662  012737  177777  001510          MOV     #-1,MEDENB
042670  016060  106314  001410  265$:   MOV     BUFFER(R0),PUTBUF(R0)
042676  062700  000002                  ADD     #2,R0           ;ADVANCE R0
042702  022700  000046                  CMP     #46,R0          ;DONE ??
042706  103370                          BHIS    265$
042710  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
042712                          300$:
042712  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042714  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042716  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042720  013737  001444  001512          MOV     RMDCO,ASNDC     ;LOAD REQUESTED CYLINDER, TRACK,
042726  013737  001416  001514          MOV     RMDAO,ASNDA     ;AND SECTOR ADDRESS IN ASSIGNED STORAGE
042734  005002                          CLR     R2              ;R2 = NUMBER OF SECTORS
042736  013700  001412                  MOV     RMWCO,R0        ;R0 = WORD COUNT
042742  005400                          NEG     R0              ;MAKE NUMBER POSITIVE
042744  012701  000400                  MOV     #256.,R1        ;R1 = NUMBER OF WORDS PER SECTOR
042750  032737  000002  001410          BIT     #BIT1,RMCS10    ;IS THIS A HEADER AND DATA COMMAND ??
042756  001402                          BEQ     305$            ;NO !!
042760  012701  000402                  MOV     #258.,R1        ;CHANGE WORDS PER SECTOR
042764  020100                  305$:   CMP     R1,R0           ;IS THERE A FULL SECTOR ??
042766  101404                          BLOS    310$            ;YES !!
042770  005700                          TST     R0              ;IS R0 ZERO ??
042772  001405                          BEQ     315$            ;YES !!
042774  005202                          INC     R2              ;INCREMENT FOR PARTIAL SECTOR
042776  000403                          BR      315$
043000  160100                  310$:   SUB     R1,R0           ;SUBTRACT ONE SECTOR FROM WORD COUNT
043002  005202                          INC     R2              ;INCREMENT SECTOR COUNT
043004  000767                          BR      305$
043006  010237  043330          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.

043012  012737  110340  043340          MOV     #MFGFIL+14,540$ ;THE STARTING ADDRESS OF MFG FILE
                                                                ;TO BASE ADDRESS STORAGE.
043020  004737  043042                  JSR     PC,320$         ;GO SEARCH FILE
043024  012737  111346  043340          MOV     #USRFIL+14,540$ ;LOAD STARTING ADDRESS OF USR FILE
                                                                ;TO BASE ADDRESS STORAGE.
043032  004737  043042                  JSR     PC,320$         ;GO SEARCH FILE
043036  000137  043302                  JMP     400$            ;DONE WITH ALL FILE SEARCHES !!

043042  013737  001512  043334  320$:   MOV     ASNDC,520$      ;LOAD COMPARING CYLINDER ADDRESS
043050  013737  001514  043336          MOV     ASNDA,530$      ;LOAD COMPARING TRACK, SECTOR ADDRESS
043056  013737  043330  043332          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
043064                                  325$:
043064  013700  043340                          MOV     540$,R0         ;LOAD THE BASE ADDRESS IN R0
043070  022710  177777                  330$:   CMP     #-1,(R0)        ;IS THIS FILE TERMINATOR ?
043074  001446                                  BEQ     370$            ;BR IF YES
043076  021037  043334                          CMP     (R0),520$       ;DOES TABLE ENTRY = COMPARING CYLINDER ?
043102  001010                                  BNE     350$            ;BR IF NO

                                        ;FILE ENTRY EQUALS COMPARING CYLINDER.  SEE IF THE NEXT ENTRY EQUALS
                                        ;THE COMPARING TRACK, AND SECTOR.
043104                                  345$:
043104  126037  000003  043337                  CMPB    3(R0),530$+1    ;DOES TABLE ENTRY = COMPARING TRACK ?
043112  001004                                  BNE     350$            ;BR IF NO

043114  126037  000002  043336                  CMPB    2(R0),530$      ;DOES TABLE ENTRY = COMPARING SECTOR ?
043122  001402                                  BEQ     360$            ;BR IF YES

043124  022020                          350$:   CMP     (R0)+,(R0)+     ;NO, ADJUST CYLINDER POINTER IN BAD FILE
043126  000760                                  BR      330$            ;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.
043130                                  360$:
043130  105237  001514                          INCB    ASNDA           ;INCREMENT SECTOR
043134  122737  000037  001514                  CMPB    #31.,ASNDA      ;SECTOR OK ??
043142  103337                                  BHIS    320$            ;YES !!
043144  105037  001514                          CLRB    ASNDA           ;CLEAR SECTOR AND ADVANCE TRACK
043150  105237  001515                          INCB    ASNDA+1
043154  123737  001333  001515                  CMPB    LSTRK+1,ASNDA+1 ;TRACK OK ?
043162  103327                                  BHIS    320$            ;YES !!
043164  005037  001514                          CLR     ASNDA           ;CLEAR TRACK AND SECTOR
043170  005237  001512                          INC     ASNDC           ;INCREMENT CYLINDER
043174  022737  001466  001512                  CMP     #822.,ASNDC     ;CYLINDER OK ??
043202  103317                                  BHIS    320$            ;YES !!
043204  005037  001512                          CLR     ASNDC           ;START AT CYLINDER 0
043210  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.
043212                                  370$:
043212  005337  043332                          DEC     510$            ;DECREMENT NUMBER OF SECTORS TO COMPARE
043216  001442                                  BEQ     410$            ;DONE IF ZERO

043220  105237  043336                          INCB    530$            ;INCREMENT THE COMPARING SECTOR
043224  122737  000037  043336                  CMPB    #31.,530$       ;SECTOR OK ??
043232  103022                                  BHIS    375$            ;YES !!
043234  105037  043336                          CLRB    530$            ;CLEAR SECTOR
043240  105237  043337                          INCB    530$+1          ;INCREMENT TRACK
043244  123737  001333  043337                  CMPB    LSTRK+1,530$+1  ;TRACK OK ??
043252  103012                                  BHIS    375$            ;YES !!
043254  005037  043336                          CLR     530$            ;CLEAR SECTOR TRACK
043260  005237  043334                          INC     520$            ;INCREMENT CYLINDER
043264  022737  001466  043334                  CMP     #822.,520$      ;CYLINDER OK ??
043272  103002                                  BHIS    375$            ;YES !!
043274  005037  043334                          CLR     520$            ;START AT CYLINDER 0
```

```
                                                ;ASSIGN THE SECTOR AND RETURN TO USER
043302                                  400$:
043302  013737  001512  001444                  MOV     ASNDC,RMDCO     ;LOAD CYLINDER
043310  013737  001514  001416                  MOV     ASNDA,RMDAO     ;LOAD TRACK AND SECTOR
043316  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
043320  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
043322  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0

043324  000240                          410$:   NOP
043326  000207                                  RTS     PC

                                                ;THE FOLLOWING ARE STORAGE LOCATIONS FOR THE MODULE

043330  000000                          500$:   .WORD   0       ;RETRY COUNT/ NUMBER OF SECTORS REQUIRED
043332  000000                          510$:   .WORD   0       ;NUMBER OF SECTORS TO COMPARE
043334  000000                          520$:   .WORD   0       ;COMPARING CYLINDER
043336  000000                          530$:   .WORD   0       ;COMPARING TRACK AND SECTOR
043340  000000                          540$:   .WORD   0       ;BASE ADDRESS OF BAD SECTOR FILE BEING SEARCHED
043342  000000                          550$:   .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

 043344                 GENBUF:
 043344  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 043346  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 043350  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 043352  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 043354  010446                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 043356  013700  001414          MOV     RMBA0,R0        ;LOAD DATA BUFFER ADDRESS
 043362  013701  001412          MOV     RMWC0,R1        ;LOAD WORD COUNT
 043366  013737  001444  043576  MOV     RMDC0,60$       ;LOAD STARTING CYLINDER ADDRESS
 043374  013737  001416  043600  MOV     RMDA0,65$       ;LOAD STARTING TRACK,SECTOR ADDRESS

 043402  032737  000002  001410 10$:    BIT     #BIT1,RMCS10    ;WRITE HEADER & DATA??
 043410  001445                 BEQ     25$             ;NO!!
 043412  013710  043576          MOV     60$,(R0)        ;WRITE HEADER WORD #1
 043416  052710  140000          BIS     #MSE!USE,(R0)   ;SET BAD SECTOR FLAGS FOR GOOD SECTOR
 043422  012702  000035          MOV     #29.,R2         ;R2 = MAXIMUM SECTOR ADDRESS (29.)

 043426  032737  010000  001442  BIT     #FMT16,RMOF0    ;18 BIT FORMAT??
 043434  001404                 BEQ     15$             ;YES !!
 043436  052710  010000          BIS     #FMT16,(R0)     ;SET 16 FORMAT BIT IN HEADER
 043442  012702  000037          MOV     #31.,R2         ;CHANGE MAXIMUM SECTOR ADDRESS (31.)

 043446  005201         15$:    INC     R1              ;INCREMENT WORD COUNT
 043450  001443                 BEQ     50$             ;EXIT IF DONE

 043452  005720                 TST     (R0)+           ;MOVE R0 TO HEADER WORD #2
 043454  013720  043600          MOV     65$,(R0)+       ;WRITE HEADER WORD #2
 043460  005201                 INC     R1              ;INCREMENT WORD COUNT AND
 043462  001436                 BEQ     50$             ;EXIT IF DONE
 043464  012703  043600          MOV     #65$,R3         ;ADVANCE SECTOR ADDRESS
 043470  105213                 INCB    (R3)
 043472  120213                 CMPB    R2,(R3)         ;SECTOR OVERFLOW ??
 043474  103013                 BHIS    25$             ;NO !!
 043476  105013                 CLRB    (R3)            ;YES - CLEAR SECTOR ADDRESS
 043500  105263  000001          INCB    1(R3)           ;ADVANCE TRACK ADDRESS
 043504  123763  001333  000001  CMPB    LSTRK+1,1(R3)   ;TRACK OVERFLOW ??
 043512  103004                 BHIS    25$             ;NO !!
 043514  105063  000001          CLRB    1(R3)           ;YES - CLEAR TRACK ADDRESS
```

```
     53 043520  105237  043576                 INCB    60$             ;ADVANCE CYLINDER ADDRESS
     54 043524  012704  000400       25$:      MOV     #256.,R4        ;LOAD SECTOR DATA COUNT
     55 043530  013702  001174       30$:      MOV     $TMP0,R2        ;LOAD PATTERN ADDRESS
     56 043534  013703  001176                 MOV     $TMP1,R3        ;LOAD PATTERN COUNT
     57 043540  012220                 40$:      MOV     (R2)+,(R0)+     ;WRITE DATA PATTERN
     58 043542  005201                           INC     R1              ;INCREMENT WORD COUNT AND
     59 043544  001405                           BEQ     50$             ;EXIT IF DONE
     60 043546  005304                           DEC     R4              ;DECREMENT SECTOR COUNT
     61 043550  001714                           BEQ     10$             ;START NEXT SECTOR IF 0
     62 043552  005303                           DEC     R3              ;DECREMENT PATTERN COUNT
     63 043554  001765                           BEQ     30$             ;RESTART PATTERN IF 0
     64 043556  000770                           BR      40$             ;CONTINUE DATA PATTERN
     65 043560                       50$:
        043560  012604                           MOV     (SP)+,R4        ;;POP STACK INTO R4
        043562  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
        043564  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
        043566  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
        043570  012600                           MOV     (SP)+,R0        ;;POP STACK INTO R0
     66 043572  000240                           NOP
     67 043574  000207                           RTS     PC
     68
     69 043576  000000                 60$:      .WORD                   ;CYLINDER ADDRESS STORAGE
     70 043600  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)    ???

 043602                         CMPBUF:
 043602  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 043604  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 043606  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 043610  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 043612  005037  044142                CLR     150$            ;CLEAR CORRECTION FLAG

                                ;DETERMINE IF DATA SHOULD BE CORRECTED
 043616  033737  004000  001366        BIT     ECI,RMOFI       ;WAS ECC CORRECTION ALLOWED ??
 043624  001063                        BNE     80$             ;NO !!
 043626  032737  100000  001350        BIT     #DCK,RMER1I     ;WAS THERE A DATA CHECK ??
 043634  001457                        BEQ     80$             ;NO !!
 043636  032737  000100  001350        BIT     #ECH,RMER1I     ;IS ERROR CORRECTION HARD SET ?
 043644  001053                        BNE     80$             ;YES !!
 043646  032737  010000  001366        BIT     #FMT16,RMOFI    ;IS THIS 16 BIT FORMAT ??
 043654  001447                        BEQ     80$             ;NO !!

                                ;CORRECT DATA USING ECC INFORMATION
 043656  013700  001414                MOV     RMBA0,R0        ;R0 = STARTING BUFFER ADDRESS
 043662  013701  001400                MOV     RMEC1I,R1       ;R1 = ECC POSITION
 043666  052737  100000  044142        BIS     #BIT15,150$     ;SET CORRECTION FLAG

                                ;MOVE R0 TO WORD BOUNDARY OF ERROR BURST
 043674  022701  000020        10$:    CMP     #16.,R1         ;IS BIT POSITION > 1 WORD
 043700  103004                        BHIS    20$             ;NO !!
 043702  162701  000020                SUB     #16.,R1         ;SUBTRACT 1 WORDS WORTH
 043706  005720                        TST     (R0)+           ;ADVANCE BUFFER ADDRESS 1 WORD
 043710  000771                        BR      10$
 043712  012702  000001        20$:    MOV     #1,R2           ;R2 = BIT POINTER
 043716  010203                        MOV     R2,R3           ;R3 = BIT NUMBER

                                ;MOVE R2 TO STARTING BIT OF ERROR BURST
 043720  020301                30$:    CMP     R3,R1           ;IS R3 SAME AS R1 ??
 043722  001403                        BEQ     35$             ;YES !!
 043724  006302                        ASL     R2              ;SHIFT BIT POINTER
 043726  005203                        INC     R3              ;INCREMENT BIT NUMBER
 043730  000773                        BR      30$
 043732  012703  000013        35$:    MOV     #11.,R3         ;R3   LENGTH OF ERROR BURST

                                ;CORRECT THE ERROR BURST
 043736  030237  001402        40$:    BIT     R2,RMEC2I       ;IS THIS BIT SET IN ECC PATTERN ??
```

```
     54 043742  001405                        BEQ     60$             ;NO - DO NOT CORRECT THIS BIT
     55 043744  030210                        BIT     R2,(R0)         ;IS THE BIT PRESENTLY SET ??
     56 043746  001402                        BEQ     50$             ;NO
     57 043750  040210                        BIC     R2,(R0)         ;RESET THE BIT
     58 043752  000401                        BR      60$
     59 043754  050210                50$:    BIS     R2,(R0)         ;SET THE BIT
     60 043756  006302                60$:    ASL     R2              ;SHIFT TO NEXT BIT
     61 043760  001003                        BNE     70$
     62 043762  012702  000001                MOV     #1,R2           ;CONTINUE WITH FIRST BIT OF NEXT WORD
     63 043766  005720                        TST     (R0)+
     64 043770  005303                70$:    DEC     R3              ;END OF BURST ??
     65 043772  001361                        BNE     40$             ;NO !!
     66
     67                                  ;COMPARE WRITE BUFFER TO READ BUFFER
     68 043774  017600  000010        80$:    MOV     @10(SP),R0      ;R0 = WRITE BUFFER
     69 044000  062766  000002  000010        ADD     #2,10(SP)       ;MOVE SP TO READ ADDRESS
     70 044006  017601  000010                MOV     @10(SP),R1      ;R1 = READ BUFFER
     71 044012  062766  000002  000010        ADD     #2,10(SP)       ;MOVE SP TO RETURN ADDRESS
     72 044020  013702  001336                MOV     RMWCI,R2        ;R2 = NUMBER OF WORDS TRANSFER
     73 044024  163702  001412                SUB     RMWCO,R2
     74 044030  022021                90$:    CMP     (R0)+,(R1)+     ;COMPARE DATA WORDS
     75 044032  001003                        BNE     100$            ;EXIT IF NOT EQUAL
     76 044034  005302                        DEC     R2              ;DECREMENT WORD COUNT
     77 044036  001374                        BNE     90$             ;CONTINUE IF NOT DONE
     78 044040  000433                        BR      110$            ;DONE COMPARE - NO ERROR
     79
     80                                  ;DATA COMPARE FAILED
     81 044042  014037  001140        100$:   MOV     -(R0),$GDDAT    ;STORE GOOD DATA FOR TYPEOUT
     82 044046  014137  001142                MOV     -(R1),$BDDAT    ;STORE BAD DATA FOR TYPEOUT
     83 044052  010037  001134                MOV     R0,$GDADR       ;STORE ADDRESS OF GOOD DATA
     84 044056  010137  001136                MOV     R1,$BDADR       ;STORE ADDRESS OF BAD DATA
     85 044062  010237  001174                MOV     R2,$TMP0        ;STORE WORD COUNT OF ERROR
     86 044066  062766  000004  000010        ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
     87 044074  112776  000336  000010        MOVB    #336,@10(SP)    ;WRITE ERROR NUMBER IN CALL
     88
     89                                  ;CHANGE ERROR NUMBER IF ECC CORRECTION FAILED
     90 044102  032737  100000  044142        BIT     #BIT15,150$     ;WAS ECC CORRECTION USED ??
     91 044110  001403                        BEQ     105$            ;NO !!
     92 044112  112776  000163  000010        MOVB    #163,@10(SP)    ;ECC CORRECTION FAILED
     93 044120  162766  000002  000010  105$:  SUB     #2,10(SP)       ;MOVE SP TO RETURN IF ERROR
     94 044126  000240                        NOP
     95 044130                        110$:
        044130  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
        044132  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
        044134  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
        044136  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
     96 044140  000207                        RTS     PC              ;RETURN TO USER
     97
     98 044142  000000                150$:   .WORD                   ;ECC CORRECTION FLAG
     99
```

```
                        .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

044144                  GETSTS:
044144  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044146  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044150  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044152  012700  001522         MOV     #GETINX,R0      ;R0 = ADDRESS OF INDEX TABLE
044156  012701  001404         MOV     #RMEC2I+2,R1    ;R1 = ADDRESS OF GET BUFFER
044162  012702  000046         MOV     #RMEC2,R2       ;R2 = REGISTER INDE
044166  110220          2$:    MOVB    R2,(R0)+        ;WRITE REGISTER INDEX IN TABLE
044170  005041                 CLR     -(R1)           ;CLEAR CORRESPONDING LOCATION
044172  162702  000002  3$:    SUB     #2,R2           ;DECREMENT TO NEXT INDEX
044176  100405                 BMI     4$              ;BRANCH OUT IF DONE
044200  022702  000022         CMP     #RMDB,R2        ;DONT WRITE RMDB INDEX
044204  001370                 BNE     2$
044206  005041                 CLR     -(R1)
044210  000770                 BR      3$
044212  112720  000200  4$:    MOVB    #200,(R0)+      ;WRITE TERMINATOR
044216  012602                 MOV     (SP)+,R2        ;;POP STACK INTO R2
044220  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
044222  012600                 MOV     (SP)+,R0        ;;POP STACK INTO R0
044224  000240                 NOP
044226  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 0  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

044230  000240         GET:    NOP
044232  062716  000004         ADD     #4,(SP)         ;CLEAR ERROR NUMBER IN USER'S
044236  105076  000000         CLRB    @(SP)           ;ERROR CALL
044242  162716  000004         SUB     #4,(SP)
044246  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044250  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044252  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044254  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
044256  010446                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
044260  013746  000004         MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
044264  013746  000006         MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
044270  013700  001276         MOV     $BASE,R0
044274  012702  001334         MOV     #GETBUF,R2
044300  012704  001522         MOV     #GETINX,R4
044304  012737  044412  000004 MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
044312  012737  000340  000006 MOV     #PR6,ERRVEC+2
044320  016037  000010  001174 1$:     MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
044326  016037  000000  001176 MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
044334  032737  004000  001176 BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
044342  001007                 BNE     3$              ;YES!!
044344  062766  000004  000016 ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN USER'S
044352  112776  000112  000016 MOVB    #112,@16(SP)    ;ERROR CALL
044360  000423                 BR      7$
044362  105714         3$:     TSTB    (R4)            ;DONE??
044364  100433                 BMI     9$              ;YES!!
044366  111401                 MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
044370  042701  177700         BIC     #^C1DXMSK,R1    ;CLEAR ANY SIGN EXTENSION
```

```
     52 044374  060001                          ADD     R0,R1
     53 044376  112403                          MOVB    (R4)+,R3        ;R3 = STORAGE ADDRESS FOR REGISTER
     54 044400  042703  177700                  BIC     #^C1DXMSK,R3    ;CLEAR ANY SIGN EXTENSION
     55 044404  060203                          ADD     R2,R3
     56 044406  011113                          MOV     (R1),(R3)       ;READ REGISTER
     57 044410  000764                          BR      3$
     58
     59 044412  022626                  5$:     CMP     (SP)+,(SP)+     ;RESTORE STACK
     60 044414  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     61 044422  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     62 044430  162766  000002  000016  7$:     SUB     #2,16(SP)
     63 044436  105714                  8$:     TSTB    (R4)            ;DONE CLEARING??
     64 044440  100405                          BMI     9$              ;YES!!
     65 044442  005003                          CLR     R3              ;CLEAR REMAINING STORAGE
     66 044444  112403                          MOVB    (R4)+,R3        ;LOCATIONS
     67 044446  060203                          ADD     R2,R3
     68 044450  005013                          CLR     (R3)
     69 044452  000771                          BR      8$
     70 044454                          9$:
        044454  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        044460  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        044464  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        044466  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        044470  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        044472  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        044474  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     71 044476  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

044500  000240          PUT:    NOP
044502  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044504  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044506  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044510  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
044512  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
044514  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
044520  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
044524  013700  001276          MOV     $BASE,R0
044530  012702  001410          MOV     #PUTBUF,R2
044534  012704  001551          MOV     #PUTINX,R4
044540  012737  044650  000004  MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
044546  012737  000300  000006  MOV     #PR6,ERRVEC+2
044554  016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET 'NED' STATUS
044562  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET 'DVA' STATUS
044570  032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
044576  001007                  BNE     3$              ;YES!!
044600  062766  000004  000016  ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
044606  112776  000112  000016  MOVB    #112,@16(SP)    ;USER'S ERROR CALL
044614  000424                  BR      7$
044616  105714          3$:     TSTB    (R4)            ;DONE??
044620  100425                  BMI     9$              ;YES!!
044622  111401                  MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
044624  042701  177700          BIC     #^C<IDXMSK>,R1  ;CLEAR ANY SIGN EXTENSION
044630  060001                  ADD     R0,R1
044632  111403                  MOVB    (R4),R3         ;R3 = STORAGE ADDRESS
044634  042703  177700          BIC     #^C<IDXMSK>,R3  ;CLEAR ANY SIGN EXTENSION
044640  060203                  ADD     R2,R3
044642  011311                  MOV     (R3),(R1)       ;WRITE REGISTER
044644  105724          4$:     TSTB    (R4)+           ;ADJUST REGISTER POINTER
```

```
     52 044646  000763                          BR      3$
     53
     54 044650  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
     55 044652  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     56 044660  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     57 044666  162766  000002  000016  7$:     SUB     #2,16(SP)
     58
     59 044674                          9$:
        044674  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        044700  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        044704  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        044706  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        044710  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        044712  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        044714  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     60 044716  000207                          RTS     PC              ;RETURN
     61
```

SEQ 0200

```
                                        .SBTTL  SIZE CLOCK SUBROUTINE
                                        
044720                                  SIZCLK:
044720  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
044724  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
044730  012737  044766  000004                  MOV     #1$,ERRVEC      ;SET UP FOR BUS TIMEOUT
044736  012737  000300  000006                  MOV     #PR6,ERRVEC+2
044744  012737  177546  001516                  MOV     #177546,CLKADR  ;LOAD ADDRESSES FOR KW11-L
044752  012737  000100  001520                  MOV     #100,CLKVCT
044760  005777  134532                          TST     @CLKADR         ;TEST FOR KW11-L PRESENT
044764  000421                                  BR      3$              ;YES - KW11-L IS PRESENT
044766  022626                          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP
044770  012737  045020  000004                  MOV     #2$,ERRVEC      ;SET UP FOR BUS TIMEOUT
044776  012737  172540  001516                  MOV     #172540,CLKADR  ;LOAD ADDRESSES FOR KW11-P CLOCK
045004  012737  000104  001520                  MOV     #104,CLKVCT
045012  005777  134500                          TST     @CLKADR         ;TEST FOR KW11-P PRESENT
045016  000404                                  BR      3$              ;YES - KW11-P IS PRESENT
045020  022626                          2$:     CMP     (SP)+,(SP)+     ;RESTORE SP
045022  062766  000002  000004                  ADD     #2,4(SP)        ;MOVE RETURN TO ERROR
045030                                  3$:
045030  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
045034  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
045040  000207                                  RTS     PC              ;RETURN TO USER
```

```
      1                                 .SBTTL  TIMEOUT SUBROUTINE
      2
      3                                 ;THIS SUBROUTINE WAITS FOR RDY = 1 AND GO = 0 OR FOR A TIMEOUT
      4                                 ;GREATER THAN APPROX. 500 MSEC AND THEN RETURNS.
      5                                 ;
      6                                 ;CALL:  JSR     PC,TIMOUT
      7                                 ;       ???                     RETURN HERE
      8                                 ;
      9 045042                          TIMOUT:
     045042  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
     045044  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
     045046  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
     045050  013746  000004                     MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
     045054  013746  000006                     MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
    10 045060  012737  045162  000004          MOV     #4$,ERRVEC      ;SETUP FOR BUS TIMEOUT - 04 TRAP
    11 045066  012737  000300  000006          MOV     #PR6,ERRVEC+2
    12 045074  013700  001276                  MOV     $BASE,R0        ;R0=BASE ADDRESS
    13 045100  013701  001516                  MOV     CLKADR,R1       ;R1=CLOCK ADDRESS
    14 045104  012702  000036                  MOV     #30.,R2         ;R2=NUMBER OF CLOCK CYCLES
    15 045110  020127  172540          1$:     CMP     R1,#172540      ;KW11-P CLOCK??
    16 045114  001003                          BNE     2$              ;NO!!
    17 045116  012761  000001  000002          MOV     #1,2(R1)        ;SET COUNTER
    18 045124  012711  000005          2$:     MOV     #BIT2!BIT0,(R1) ;START COUNTER
    19
    20 045130  016046  000000          3$:     MOV     RMCS1(R0),-(SP) ;GET STATUS
    21 045134  042716  177576                  BIC     #^C<RDY!GO>,(SP)
    22 045140  022726  000200                  CMP     #RDY,(SP)+      ;RDY=1,GO=0??
    23 045144  001420                          BEQ     5$              ;YES!!
    24 045146  032711  000200                  BIT     #BIT7,(R1)      ;TIMER DONE??
    25 045152  001766                          BEQ     3$              ;NO!!
    26 045154  005302                          DEC     R2              ;DEC NUMBER OF CYCLES
    27 045156  001354                          BNE     1$              ;CONTINUE IF NOT DONE
    28 045160  000412                          BR      5$              ;'RDY' DID NOT SET OR 'GO' DID NOT RESET
    29                                                                  ;WITHIN 500 MSEC AFTER THE COMMAND WAS ISSUED.
    30 045162  022626                  4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
    31 045164  062766  000004  000012          ADD     #4,12(SP)       ;MOVE SP TO USER'S CALL
    32 045172  112776  000007  000012          MOVB    #7,@12(SP)      ;WRITE ERROR NUMBER
    33 045200  162766  000002  000012          SUB     #2,12(SP)
    34
    35 045206                          5$:
     045206  012637  000006                     MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
     045212  012637  000004                     MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     045216  012602                             MOV     (SP)+,R2        ;;POP STACK INTO R2
     045220  012601                             MOV     (SP)+,R1        ;;POP STACK INTO R1
     045222  012600                             MOV     (SP)+,R0        ;;POP STACK INTO R0
    36 045224  000207                          RTS     PC              ;RETURN TO USER
    37
```

```
                                .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 PMD, HAVE BEEN STORED IN THEIR
                                ;CORRESPONDING LOCATIONS OF THE 'GE.' 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

045226                          PRIERR:

                                ;CLEAR USER'S ERROR CALL
045226  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
045232  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
045236  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

                                ;REPORT AN ERROR IF THE WRONG UNIT IS SELECTED
045242  013737  001344  001142          MOV     RMCS2I,$BDDAT   ;CORRECT UNIT SELECTED??
045250  042737  177770  001142          BIC     #^CUNTMSK,$BDDAT
045256  013737  001234  001140          MOV     $UNIT,$GDDAT    ;GOOD DATA FOR TYPEOUT
045264  042737  177770  001140          BIC     #^CUNTMSK,$GDDAT
045272  123737  001140  001142          CMPB    $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                                                ;DRIVE NUMBERS
045300  001415                          BEQ     1$              ;YES!!
045302  062716  000004                  ADD     #4,(SP)
045306  112776  000001  000000          MOVB    #1,@(SP)        ;ERROR 1
045314  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045320  004736                          JSR     PC,@(SP)+       ;REPORT WRONG UNIT SELECTED
```

```
     58 045322  162716  000010                 SUB     #10,(SP)        ;RESTORE (SP)
     59 045326  000240                          NOP
     60 045330  000137  046050                 JMP     10$             ;SKIP OTHER CHECKS
     61 045334                          1$:
     62
     63                                 ;REPORT AN ERROR IF THE DEVICE IS NOT AVAILABLE OR IF
     64                                 ;THE DEVICE IS NONEXISTANT
     65 045334  032737  004000  001334          BIT     #DVA,RMCS1I     ;DEVICE AVAILABLE??
     66 045342  001045                          BNE     5$              ;YES!!
     67 045344  013737  001334  001140          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
     68 045352  052737  004000  001140          BIS     #DVA,$GDDAT
     69 045360  013737  001334  001142          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
     70 045366  062716  000004                  ADD     #4,(SP)
     71 045372  112776  000002  000000          MOVB    #2,@(SP)        ;ERROR #2
     72 045400  032737  010000  001344          BIT     #NED,RMCS2I     ;WAS NED SET??
     73 045406  001414                          BEQ     2$              ;NO!!
     74 045410  013737  001344  001140          MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
     75 045416  013737  001344  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
     76 045424  042737  010000  001140          BIC     #NED,$GDDAT
     77 045432  112776  000003  000000          MOVB    #3,@(SP)        ;YES - CHANGE ERROR NUMBER
     78 045440  162716  000002          2$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     79 045444  004736                          JSR     PC,@(SP)+       ;REPORT DEVICE NOT AVAILABLE
     80 045446  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
     81 045452  000240                          NOP
     82 045454  000575                          BR      10$             ;SKIP OTHER CHECKS
     83 045456                          5$:
     84
     85                                 ;REPORT AN ERROR IF MASSBUS CONTROLLER IS NOT READY
     86 045456  032737  000200  001334          BIT     #RDY,RMCS1I     ;CONTROLLER READY??
     87 045464  001030                          BNE     7$              ;YES!!
     88 045466  013737  001334  001140          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
     89 045474  052737  000200  001140          BIS     #RDY,$GDDAT
     90 045502  042737  160001  001140          BIC     #SC!TRE!MCPE!GO,$GDDAT
     91 045510  013737  001334  001142          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
     92 045516  062716  000004                  ADD     #4,(SP)
     93 045522  112776  000004  000000          MOVB    #4,@(SP)        ;ERROR #4
     94 045530  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     95 045534  004736                          JSR     PC,@(SP)+       ;REPORT CONTROLLER NOT READY
     96 045536  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
     97 045542  000240                          NOP
     98 045544  000541                          BR      10$             ;SKIP OTHER CHECKS
     99 045546                          7$:
    100
    101                                 ;REPORT AN ERROR IF GO IS NOT ZERO AND DRY IS NOT ONE
    102 045546  032737  000001  001334          BIT     #GO,RMCS1I      ;GO RESET??
    103 045554  001431                          BEQ     8$              ;YES!!
    104 045556  032737  000200  001346          BIT     #DRY,RMDSI      ;DRIVE READY??
    105 045564  001025                          BNE     8$              ;YES!!
    106 045566  013737  001334  001140          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
    107 045574  042737  160001  001140          BIC     #SC!TRE!MCPE!GO,$GDDAT
    108 045602  013737  001334  001142          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
    109 045610  062716  000004                  ADD     #4,(SP)
    110 045614  112776  000005  000000          MOVB    #5,@(SP)        ;ERROR #5
    111 045622  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    112 045626  004736                          JSR     PC,@(SP)+       ;REPORT DRIVE NOT READY
    113 045630  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
    114 045634  000240                          NOP
```

```
                                        BR      10$
                                8$:
                                ;REPORT AN ERROR IF THE RH CONTROLLER DETECTED BAD
                                ;PARITY ON THE MASSBUS CONTROL BUS
045640  032737  020000  001334            BIT     #MCPE,RMCS1I    ;PARITY ERROR ??
045646  001425                            BEQ     9$              ;NO!!
045650  013737  001334  001140            MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
045656  042737  160001  001140            BIC     #SC!TRE!MCPE!GO,$GDDAT
045664  013737  001334  001142            MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
045672  062716  000004                    ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
045676  112776  000013  000000            MOVB    #13,@(SP)       ;ERROR #13
045704  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045710  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
045712  162716  000010                    SUB     #10,(SP)        ;RESTORE STACK
045716  000240                            NOP
045720  000453                            BR      10$
045722                          9$:

                                ;REPORT AN ERROR IF DETECTED A CONTROL BUS PARITY ERROR
045722  032737  000010  001350            BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
045730  001451                            BEQ     11$             ;NO!!
045732  032737  000010  001376            BIT     #DPE,RMER2I     ;WAS IT THE CONTROL BUS??
045740  001045                            BNE     11$             ;NOT SURE!!
045742  032737  000010  001424            BIT     #PAR,RMER1O     ;DID TEST SET PAR ??
045750  001413                            BEQ     93$             ;NO!!
045752  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
045754  012700  001551                    MOV     #PUTINX,R0      ;R0 POINTS TO INDEX TABLE
045760  122710  000014          91$:      CMPB    #RMER1,(R0)     ;SEARCH TABLE FOR RMER1
045764  001002                            BNE     92$
045766  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
045770  000431                            BR      11$             ;PAR WAS SET BY TEST
045772  105720                  92$:      TSTB    (R0)+           ;END OF TABLE??
045774  100371                            BPL     91$             ;NO!!
045776  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
046000  013737  001350  001140  93$:      MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
046006  042737  000010  001140            BIC     #PAR,$GDDAT
046014  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
046022  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
046026  112776  000050  000000            MOVB    #50,@(SP)       ;WRITE THE ERROR NUMBER
046034  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046040  004736                            JSR     PC,@(SP)+       ;REPORT THE ERROR
046042  162716  000010                    SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
046046  000240                            NOP
046050  062716  000010          10$:      ADD     #10,(SP)        ;RETURN TO ERROR
046054  000240                  11$:      NOP                     ;RETURN TO NO ERROR
046056  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.
                                
046060                          SECERR:
                                
                                ;;*****************************************************************
                                ;STORE FUNCTION CODE AND CLEAR USER'S ERROR NUMBER
046060  013737  001410  051714          MOV     RMCS10,515$     ;STORE FUNCTION CODE
046066  042737  177701  051714          BIC     #^C<F0!F1!F2!F3!F4>,515$
046074  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
046100  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
046104  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
                                
                                ;;*****************************************************************
                                ;CHECK SECONDARY ERRORS COMMON TO ALL COMMANDS
                                
                                ;REPORT ERROR IF DRIVE IS NOT READY, I.E., IF DRY = 0
046110  032737  000200  001346          BIT     #DRY,RMDSI      ;DRIVE READY??
046116  001024                          BNE     5$              ;YES!!
046120  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
046126  042737  177577  001142          BIC     #^CDRY,$BDDAT
046134  012737  000200  001140          MOV     #DRY,$GDDAT     ;GOOD DATA FOR TYPEOUT
046142  062716  000004                  ADD     #4,(SP)
046146  112776  000010  000000          MOVB    #10,@(SP)       ;ERROR NUMBER
046154  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046160  004736                          JSR     PC,@(SP)+       ;REPORT NOT READY
046162  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP) TO ERROR N
046166  000240                          NOP
                                
                                ;REPORT ERROR IF GO BIT IS NOT RESET
046170  032737  000001  001334  5$:     BIT     #GO,RMCS1I      ;GO BIT RESET??
046176  001423                          BEQ     10$             ;YES!!
046200  013737  001334  001142          MOV     RMCS1I,$BDDAT   ;BAD DATA FOR TYPEOUT
046206  042737  177776  001142          BIC     #^CGO,$BDDAT
046214  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
046220  062716  000004                  ADD     #4,(SP)
046224  112776  000011  000000          MOVB    #11,@(SP)       ;ERROR NUMBER
```

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

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

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

```
    115 046534  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
    116 046540  000240                          NOP
    117 046542                          45$:
    118
    119                                 ;;*****************************************************************
    120                                 ;USING THE FUNCTION CODE TABLE, CHECK FOR THE FOLLOWING ERRORS:
    121                                 ;       .STATUS BITS NOT SET THAT SHOULD BE SET, E.G., ATA AND ILF
    122                                 ;       .STATUS BITS SET THAT SHOULD NOT BE SET, E.G., WCE AND ECH
    123                                 ;NOTE THAT SOME ERROR BITS ARE CONDITIONAL ON THE COMMAND AND OTHER
    124                                 ;STATUS CONDITIONS, E.G., WRITE LOCK ERROR SHOULD ONLY BE SET IF
    125                                 ;WRITE LOCK IS ON AND THF COMMAND IS A WRITE.
    126
    127                                 ;GET AND STORE THE ENTRY FROM THE FUNCTION CODE TABLE
    128 046542  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
    129 046544  013700  051714                  MOV     515$,R0         ;GET FUNCTION CODE
    130 046550  016037  071516  051706          MOV     FNCDTB(R0),500$ ;STORE ENTRY
    131 046556  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
    132
    133                                 ;REPORT AN ERROR IF AN UNEXPECTED ATTENTION OCCURRED OR IF
    134                                 ;ATA IS NOT SET AND SHOULD BE SET.
    135 046560  013737  051706  001140          MOV     500$,$GDDAT     ;GET EXPECTED ATA STATUS
    136 046566  032737  040000  001346          BIT     #ERR,RMDSI      ;IS COMPOSITE ERROR SET ??
    137 046574  001403                          BEQ     50$             ;NO !!
    138 046576  052737  100000  001140          BIS     #ATA,$GDDAT     ;EXPECT AN ATTENTION
    139 046604  042737  077777  001140  50$:    BIC     #^CATA,$GDDAT   ;STRIP DONT CARES
    140 046612  013737  001346  001142          MOV     RMDSI,$BDDAT    ;GET RECEIVED ATA
    141 046620  042737  077777  001142          BIC     #^CATA,$BDDAT   ;STRIP DONT CARES
    142 046626  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA OK ??
    143 046634  001413                          BEQ     55$             ;YES !!
    144 046636  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    145 046642  112776  000006  000000          MOVB    #6,@(SP)        ;LOAD ERROR # IN CALL
    146 046650  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    147 046654  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
    148 046656  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
    149 046662  000240                          NOP
    150 046664                          55$:
    151
    152                                 ;REPORT ERROR IF ILF IS INCORRECT, I.E., IF ILF DOES NOT COMPARE
    153                                 ;WITH FUNCTION CODE TABLE
    154 046664  013737  051706  001140          MOV     500$,$GDDAT     ;GET EXPECTED ILF
    155 046672  042737  177776  001140          BIC     #^CILF,$GDDAT   ;CLEAR ALL OTHER BITS
    156 046700  013737  001350  001142          MOV     RMER1I,$BDDAT   ;GET RECEIVED ILF
    157 046706  042737  177776  001142          BIC     #^CILF,$BDDAT   ;CLEAR ALL OTHER BITS
    158 046714  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ILF OK ??
    159 046722  001412                          BEQ     60$             ;YES !!
    160 046724  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    161 046730  112776  000254  000000          MOVB    #254,@(SP)      ;WRITE ERROR NUMBER IN CALL
    162 046736  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    163 046742  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    164 046744  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    165 046750  005037  001140          60$:    CLR     $GDDAT          ;CLEAR EXPECTED STATUS
    166
    167                                 ;REPORT AN ERROR IF WCE IS SET AND SHOULD NOT BE SET
    168 046754  013746  051706                  MOV     500$,-(SP)      ;GET WCE STATUS ENABLE
    169 046760  052716  137777                  BIS     #^CWCE,(SP)     ;SET ALL OTHER BITS
    170 046764  013737  001344  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
    171 046772  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR WCE IF ENABLED
```

```
047776  001412                          BEQ     90$             ;BRANCH IF WCE OK
047000  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
047004  112776  000026  000000          MOVB    #26,@(SP)       ;WRITE ERROR NUMBER
047012  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
047016  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
047020  162716  000010                  SUB     #10,(SP)        ;RESTORE ERROR
047024                          90$:

                                ;REPORT ERROR IF OPI STATUS IS SET AND SHOULD NOT BE SET
047024  013746  051706                  MOV     500$,-(SP)      ;GET OPI STATUS ENABLE
047030  052716  157777                  BIS     #^COPI,(SP)     ;SET ALL OTHER BITS
047034  013737  001350  001142          MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
047042  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR OPI IF ENABLED
047046  001412                          BEQ     100$            ;BRANCH IF OPI OK
047050  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
047054  112776  000164  000000          MOVB    #164,@(SP)      ;WRITE ERROR NUMBER IN CALL
047062  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
047066  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
047070  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
047074                          100$:

                                ;REPORT ERROR IF IVC IS SET AND IS NOT ENABLED OR IF IVC IS
                                ;SET AND VV IS NOT RESET
047074  013746  051706                  MOV     500$,-(SP)      ;GET IVC STATUS ENABLE
047100  032737  000100  001346          BIT     #VV,RMDSI       ;IS VV SET
047106  001402                          BEQ     105$            ;NO !!
047110  042716  010000                  BIC     #IVC,(SP)       ;YES - IVC SHOULD BE 0
047114  052716  167777          105$:   BIS     #^CIVC,(SP)     ;SET ALL OTHER BITS
047120  013737  001376  001142          MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
047126  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR IVC IF ENABLED
047132  001412                          BEQ     110$            ;BRANCH IF IVC OK
047134  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
047140  112776  000165  000000          MOVB    #165,@(SP)      ;WRITE ERROR NUMBER IN CALL
047146  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
047152  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
047154  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
047160                          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
047160  012746  177777                  MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
047164  032737  004000  051706          BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
047172  001404                          BEQ     115$            ;NO !!
047174  032737  004000  001346          BIT     #WRL,RMDSI      ;IS THE DRIVE WRITE PROTECTED ??
047202  001002                          BNE     120$            ;YES !!
047204  042716  004000          115$:   BIC     #WLE,(SP)       ;RESET WLE ENABLE
047210  013737  001350  001142  120$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
047216  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR WLE IF ENABLED
047222  001412                          BEQ     125$            ;BRANCH IF WLE OK
```

```
    229 047224  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USERS ERROR CALL
    230 047230  112776  000023  000000          MOVB    #23,@(SP)        ;WRITE ERROR NUMBER IN CALL
    231 047236  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO ERROR RETURN
    232 047242  004736                          JSR     PC,@(SP)+        ;REPORT ERROR AND RETURN
    233 047244  162716  000010                  SUB     #10,(SP)         ;RESTORE SP TO NO ERROR
    234 047250                          125$:
    235
    236                                 ;REPORT ERROR IF WCF IS SET AND WRITE ERRORS ARE NOT ENABLED
    237 047250  012746  177777                  MOV     #-1,-(SP)        ;ASSUME WRITE ERRORS ENABLED
    238 047254  032737  004000  051706          BIT     #WLE,500$        ;ARE WRITE ERRORS ENABLED ??
    239 047262  001002                          BNE     130$             ;YES !!
    240 047264  042716  000040                  BIC     #WCF,(SP)        ;DISABLE WCF ERROR
    241 047270  013737  001350  001142  130$:   MOV     RMER1I,$BDDAT    ;GET RECEIVED STATUS
    242 047276  042637  001142                  BIC     (SP)+,$BDDAT     ;RESET WCF IF ENABLED
    243 047302  001412                          BEQ     135$             ;BRANCH IF WCF OK
    244 047304  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USERS ERROR CALL
    245 047310  112776  000025  000000          MOVB    #25,@(SP)        ;WRITE ERROR NUMBER IN CALL
    246 047316  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO ERROR RETURN
    247 047322  004736                          JSR     PC,@(SP)+        ;REPORT ERROR
    248 047324  162716  000010                  SUB     #10,(SP)         ;RESTORE SP TO NO ERROR
    249 047330                          135$:
    250
    251                                 ;REPORT ERROR IF DPE IS SET AND WRITE ERRORS ARE NOT ENABLED
    252 047330  012746  177777                  MOV     #-1,-(SP)        ;ASSUME WRITE ERRORS ARE ENABLED
    253 047334  032737  004000  051706          BIT     #WLE,500$        ;ARE WRITE ERRORS ENABLED ??
    254 047342  001002                          BNE     140$             ;YES !!
    255 047344  042716  000010                  BIC     #DPE,(SP)        ;RESET DPE ENABLE
    256 047350  013737  001376  001142  140$:   MOV     RMER2I,$BDDAT    ;GET RECEIVED STATUS
    257 047356  042637  001142                  BIC     (SP)+,$BDDAT     ;RESET DPE IF ENABLED
    258 047362  001412                          BEQ     145$             ;BRANCH IF DPE OK
    259 047364  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USERS ERROR CALL
    260 047370  112776  000040  000000          MOVB    #40,@(SP)        ;WRITE ERROR NUMBER IN CALL
    261 047376  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO ERROR RETURN
    262 047402  004736                          JSR     PC,@(SP)+        ;REPORT ERROR
    263 047404  162716  000010                  SUB     #10,(SP)         ;RESTORE SP TO NO ERROR
    264 047410                          145$:
    265
    266                                 ;REPORT AN ERROR IF UPE IS SET AND WRITE ERRORS ARE NOT ENABLED
    267 047410  012746  177777                  MOV     #-1,-(SP)        ;ASSUME WRITE ERRORS ARE ENABLED
    268 047414  032737  004000  051706          BIT     #WLE,500$        ;ARE WRITE ERRORS ENABLED ??
    269 047422  001002                          BNE     150$             ;YES !!
    270 047424  042716  020000                  BIC     #UPE,(SP)        ;DISABLE UPE ERROR
    271 047430  013737  001344  001142  150$:   MOV     RMCS2I,$BDDAT    ;GET RECEIVED STATUS
    272 047436  042637  001142                  BIC     (SP)+,$BDDAT     ;RESET UPE IF ENABLED
    273 047442  001412                          BEQ     155$             ;BRANCH IF UPE OK
    274 047444  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USERS ERROR CALL
    275 047450  112776  000024  000000          MOVB    #24,@(SP)        ;WRITE ERROR NUMBER IN CALL
    276 047456  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO ERROR RETURN
    277 047462  004736                          JSR     PC,@(SP)+        ;REPORT ERROR AND RETURN
    278 047464  162716  000010                  SUB     #10,(SP)         ;MOVE SP TO NO ERROR
    279 047470                          155$:
    280
    281                                 ;REPORT AN ERROR IF IAE IS SET AND IS NOT ENABLED
    282 047470  013746  051706                  MOV     500$,-(SP)       ;GET IAE ENABLE
    283 047474  052716  175777                  BIS     #^CIAE,(SP)      ;SET ALL OTHER BITS
    284 047500  013737  001350  001142          MOV     RMER1I,$BDDAT    ;GET RECEIVED STATUS
    285 047506  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 !!
```

```
343 047714 042716 001000                BIC     #MXF,(SP)       ;DISABLE MXF ERROR
344 047720 013737 001344 001142 185$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
345 047726 042637 001142                BIC     (SP)+,$BDDAT    ;CLEAR MXF IF ENABLED
346 047732 001412                       BEQ     190$            ;BRANCH IF MXF IS OK
347 047734 062716 000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
348 047740 112776 000033 000000         MOVB    #33,@(SP)       ;WRITE ERROR NUMBER IN CALL
349 047746 162716 000002                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
350 047752 004736                       JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
351 047754 162716 000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
352 047760                      190$:
353
354                             ;REPORT ERROR IF AOE IS SET AND DATA ERRORS ARE NOT ENABLED
355 047760 012746 177777                MOV     #-1,-(SP)       ;ASSUME DATA ERRORS ARE ENABLED
356 047764 032737 001000 051706         BIT     #AOE,500$       ;ARE DATA ERRORS EANBLED ??
357 047772 001404                       BEQ     191$            ;NO !!
358 047774 032737 002000 001346         BIT     #LBT,RMDSI      ;IS LBT ALSO SET ??
359 050002 001002                       BNE     195$            ;YES !!
360 050004 042716 001000        191$:   BIC     #AOE,(SP)       ;DISABLE AOE
361 050010 013737 001350 001142 195$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
362 050016 042637 001142                BIC     (SP)+,$BDDAT    ;CLEAR AOE IF ENABLED
363 050022 001412                       BEQ     200$            ;BRANCH IF AOE IS OK
364 050024 062716 000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
365 050030 112776 000020 000000         MOVB    #20,@(SP)       ;WRITE ERROR NUMBER
366 050036 162716 000002                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
367 050042 004736                       JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
368 050044 162716 000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
369 050050                      200$:
370
371                             ;BIT 07 (HCE) OF THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR
372                             ;HEADER ERRORS, I.E.,
373                             ;       RMER1 - HCRC,HCE,FER
374                             ;       RMER2 - BSE
375
376                             ;RESET THE ENABLING BIT (HCE) IF HEADER COMPARE INHIBIT IS SET
377 050050 032737 002000 001366          BIT     #HCI,RMOFI      ;IS HCI SET ??
378 050056 001403                       BEQ     201$            ;NO !!
379 050060 042737 000200 051706         BIC     #HCE,500$       ;YES - DISABLE ALL HEADER ERRORS
380 050066                      201$:
381
382                             ;REPORT AN ERROR IF HCRC IS SET AND HEADER ERRORS ARE NOT ENABLED
383 050066 012746 177777                MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
384 050072 032737 000200 051706         BIT     #HCE,500$       ;ARE HEADER ERRORS ENABLED ??
385 050100 001002                       BNE     205$            ;YES !!
386 050102 042716 000400                BIC     #HCRC,(SP)      ;DISABLE HCRC
387 050106 013737 001350 001142 205$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
388 050114 042637 001142                BIC     (SP)+,$BDDAT    ;RESET HCRC IF ENABLED
389 050120 001412                       BEQ     210$            ;BRANCH IF HCRC IS OK
390 050122 062716 000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
391 050126 112776 000035 000000         MOVB    #35,@(SP)       ;WRITE ERROR NUMBER IN CALL
392 050134 162716 000002                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
393 050140 004736                       JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
394 050142 162716 000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
395 050146                      210$:
396
397                             ;REPORT ERROR IF HCE IS SET AND HEADER ERRORS ARE NOT ENABLED
398 050146 012746 177777                MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
399 050152 032737 000200 051706         BIT     #HCE,500$       ;ARE ERRORS ENABLED ??
```

```
    400 050160  001002                          BNE     215$            ;YES !!
    401 050162  042716  000200                  BIC     #HCE,(SP)       ;DISABLE HCE
    402 050166  013737  001350  001142  215$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
    403 050174  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR HCE IF ENABLED
    404 050200  001412                          BEQ     220$            ;BRANCH IF HCE IS OK
    405 050202  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    406 050206  112776  000036  000000          MOVB    #36,@(SP)       ;WRITE ERROR NUMBER IN CALL
    407 050214  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    408 050220  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    409 050222  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    410 050226                          220$:
    411
    412                                 ;REPORT ERROR IF FER IS SET AND HEADER ERRORS ARE NOT ENABLED
    413 050226  012746  177777                  MOV     #-1,-(SP)       ;ASSUME FER IS ENABLED
    414 050232  032737  000200  051706          BIT     #HCE,500$       ;ARE HEADER ERRORS ENABLED ??
    415 050240  001002                          BNE     225$            ;YES !!
    416 050242  042716  000020                  BIC     #FER,(SP)       ;DISABLE FER
    417 050246  013737  001350  001142  225$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
    418 050254  042637  001142                  BIC     (SP)+,$BDDAT    ;RESET FER IF ENABLED
    419 050260  001412                          BEQ     230$            ;BRANCH IF FER OK
    420 050262  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    421 050266  112776  000037  000000          MOVB    #37,@(SP)       ;WRITE ERROR NUMBER IN CALL
    422 050274  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    423 050300  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    424 050302  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    425 050306                          230$:
    426
    427                                 ;REPORT ERROR IF BSE IS SET AND HEADER ERRORS ARE NOT ENABLED
    428 050306  012746  177777                  MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
    429 050312  032737  000200  051706          BIT     #HCE,500$       ;ARE THEY ENABLED ??
    430 050320  001002                          BNE     235$            ;YES !!
    431 050322  042716  100000                  BIC     #BSE,(SP)       ;DISABLE BSE
    432 050326  013737  001376  001142  235$:   MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
    433 050334  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR BSE IF ENABLED
    434 050340  001412                          BEQ     240$            ;BRANCH IF BSE OK
    435 050342  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    436 050346  112776  000354  000000          MOVB    #354,@(SP)      ;WRITE ERROR NUMBER
    437 050354  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    438 050360  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    439 050362  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    440 050366                          240$:
    441
    442                                 ;BIT 06 OF THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR DATA
    443                                 ;FIELD ERRORS, I.E.,
    444                                 ;       RMCS2 - MDPE
    445                                 ;       RMER1 - DCK,ECH
    446                                 ;NOTE:
    447                                 ;       ECH CANNOT SET UNLESS IT IS ENABLED AND ECI IS RESET AND
    448                                 ;DCK IS SET.
    449
    450                                 ;REPORT ERROR IF MDPE IS SET AND IS NOT ENABLED
    451 050366  012746  177777                  MOV     #-1,-(SP)       ;ASSUME ENABLED
    452 050372  032737  000100  051706          BIT     #ECH,500$       ;ARE DATA FIELD ERRORS ENABLED ??
    453 050400  001002                          BNE     245$            ;YES !!
    454 050402  042716  000400                  BIC     #MDPE,(SP)      ;DISBALE MDPE
    455 050406  013737  001344  001142  245$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
    456 050414  042637  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$:   MOV     RMER1I,$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,RMOFI      ;IS ECI SET ??
                                                BNE     265$            ;YES !!
                                                BIT     #DCK,RMER1I     ;IS DCK ALSO SET ??
                                                BNE     270$            ;YES !!
                                        265$:   BIC     #ECH,(SP)       ;DISABLE ECH
                                        270$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$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
                        ;;*****************************************************************

050626  022737  000030  051714        CMP     #SEARCH,515$    ;WAS DATA TRANSFERRED ?
050634  103402                        BLO     280$            ;BR IF YES
050636  000137  051660                JMP     355$            ;NO - EXIT

                        ;REPORT ERROR IF RMWC NOT ZERO AND TRE IS ZERO
050642  013737  001336  001142  280$:  MOV     RMWCI,$BDDAT    ;WORD COUNT ZERO??
050650  001421                        BEQ     285$            ;YES
050652  032737  040000  001334        BIT     #TRE,RMCS1I     ;TRANSFER ERROR DETECTED??
050660  001015                        BNE     285$            ;YES!!
050662  062716  000004                ADD     #4,(SP)
050666  112776  000015  000000        MOVB    #15,@(SP)       ;ERROR NUMBER
050674  005037  001140                CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
050700  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050704  004736                        JSR     PC,@(SP)+       ;REPORT WORD COUNT NOT ZERO
050706  162716  000010                SUB     #10,(SP)        ;RESTORE (SP)
050712  000240                        NOP

                        ;REPORT ERROR IF RMBA IS NOT CORRECT
050714  013737  001336  001140  285$:  MOV     RMWCI,$GDDAT    ;GET WORD COUNT AT END OF TRANSFER AND
050722  163737  001412  001140        SUB     RMWCO,$GDDAT    ;SUBTRACT STARTING WORD COUNT.
050730  006337  001140                ASL     $GDDAT          ;* 2
050734  063737  001414  001140        ADD     RMBAO,$GDDAT    ;ADD STARTING BUS ADDRESS

050742  032737  000010  001344        BIT     #BAI,RMCS2I     ;WAS BUS ADDRESS INHIBIT (BAI) SET ??
050750  001403                        BEQ     290$            ;NO !!
050752  013737  001414  001140        MOV     RMBAO,$GDDAT    ;ADDRESS SHOULD NOT HAVE CHANGED

050760  023737  001140  001340  290$:  CMP     $GDDAT,RMBAI    ;BUS ADDRESS OK??
050766  001416                        BEQ     295$            ;YES!!
050770  013737  001340  001142        MOV     RMBAI,$BDDAT    ;BAD DATA FOR TYPEOUT
050776  062716  000004                ADD     #4,(SP)
051002  112776  000016  000000        MOVB    #16,@(SP)       ;ERROR NUMBER
051010  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051014  004736                        JSR     PC,@(SP)+       ;REPORT UNEXPECTED ADDRESS
051016  162716  000010                SUB     #10,(SP)        ;RESTORE (SP)
051022  000240                        NOP

                        ;COMPUTE NUMBER OF SECTORS TRANSFERRED FROM WORD COUNT
051024  005046                295$:   CLR     -(SP)           ;NUMBER OF SECTORS TRANSFERRED
051026  013746  001336                MOV     RMWCI,-(SP)     ;GET WORD COUNT AT END OF TRANSFER AND
051032  163716  001412                SUB     RMWCO,(SP)      ;SUBTRACT STARTING WORD COUNT.

051036  012746  000400                MOV     #256.,-(SP)     ;ASSUME 256. WORDS PER SECTOR
051042  032737  000002  001410        BIT     #BIT1,RMCS1O    ;HEADER & DATA COMMAND ??
051050  001402                        BEQ     300$            ;NO !!
051052  062716  000002                ADD     #2,(SP)         ;CHANGE TO 258. WORDS PER SECTOR

051056  005266  000004        300$:   INC     4(SP)           ;INCREMENT SECTOR COUNT
051062  161666  000002                SUB     (SP),2(SP)      ;SUBTRACT ONE SECTOR'S WORTH
051066  003373                        BGT     300$            ;CONTINUE IF NOT DONE
051070  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK

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

```
                                        ;NUMBER OF SECTORS
051072  013737  001444  051706        MOV     RMDCO,500$      ;STORE ORIGINAL CYLINDER
051100  013737  001416  051710        MOV     RMDAO,505$      ;STORE ORIGINAL TRACK
051106  013737  001416  051712        MOV     RMDAO,510$      ;STORE ORIGINAL SECTOR
051114  013737  001332  051716        MOV     LSTRK,520$      ;STORE LAST TRACK,
051122  000337  051716                SWAB    520$            ;GET TRACK ADDRESS TO LO BYTE AND
051126  005237  051716                INC     520$            ;INCREMENT TO GET TOTL # OF TRACKS.

051132  042737  000377  051710        BIC     #^C<TADMSK>,505$        ;SAVE TRACK ADDRESS BITS AND
051140  000337  051710                SWAB    505$            ;SWAP TRACK ADDRESS TO LOW BYTE.
051144  042737  177400  051712        BIC     #^C<SADMSK>,510$        ;SAVE SECTOR ADDRESS BITS
051152  062637  051712                ADD     (SP)+,510$

051156  023727  051712  000040  310$: CMP     510$,#32.       ;SECTOR OVEFLOWED??
051164  103406                        BLO     315$            ;NO!!
051166  005237  051710                INC     505$            ;INCREMENT TRACK
051172  162737  000040  051712        SUB     #32.,510$       ;ADJUST SECTOR
051200  000766                        BR      310$            ;TRY AGAIN

051202  023737  051710  051716  315$: CMP     505$,520$       ;TRACK OVERFLOWED??
051210  103407                        BLO     320$            ;NO!!
051212  005237  051706                INC     500$            ;INCREMENT CYLINDER
051216  163737  051716  051710        SUB     520$,505$       ;ADJUST TRACK
051224  000766                        BR      315$            ;TRY AGAIN
051226  000240                        NOP

                                        ;REPORT ERROR IF "LBT" IS NOT CORRECT
051230                                  320$:
051230  005037  001140                CLR     $GDDAT          ;SET GOOD DATA FOR LBT = 0
051234  023727  051706  001466        CMP     500$,#822.      ;SHOULD LBT BE SET??
051242  101407                        BLOS    325$            ;NO!!
051244  032737  002000  001350        BIT     #IAE,RMER1I     ;WAS IAE SET ??
051252  001003                        BNE     325$            ;YES - LBT SHOULD NOT BE SET
051254  012737  002000  001140        MOV     #LBT,$GDDAT     ;SET GOOD DATA FOR LBT = 1
051262  013737  001346  001142  325$: MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
051270  042737  175777  001142        BIC     #^CLBT,$BDDAT
051276  023737  001140  001142        CMP     $GDDAT,$BDDAT   ;IS LBT CORRECT??
051304  001413                        BEQ     330$            ;YES!!
051306  062716  000004                ADD     #4,(SP)
051312  112776  000017  000000        MOVB    #17,@(SP)       ;ERROR NUMBER
051320  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051324  004736                        JSR     PC,@(SP)+       ;REPORT LBT IS WRONG
051326  162716  000010                SUB     #10,(SP)        ;RESTORE (SP)
051332  000240                        NOP

                                        ;REPORT ERROR IF "AOE" IS INCORRECT
051334  005037  001140                330$:   CLR     $GDDAT          ;SET FOR AOE = 0
051340  032737  002000  001350        BIT     #IAE,RMER1I     ;WAS "IAE" DETECTED??
051346  001031                        BNE     340$            ;YES-"AOE" SHOULD BE ZERO
051350  023727  051706  001466        CMP     500$,#822.      ;SHOULD AOE BE SET??
051356  101425                        BLOS    340$            ;NO!!
051360  005737  051710                TST     505$            ;MAYBE
051364  001012                        BNE     335$            ;YES
051366  005737  051712                TST     510$
051372  001007                        BNE     335$            ;YES !!
051374  032737  000010  051714        BIT     #F2,515$        ;WAS THIS READ OR WRITE CHECK ??
051402  001413                        BEQ     340$            ;NO !!
```

```
    115 051404  005737  001336                    TST     RMWCI           ;WAS ALL DATA TRANSFERRED ??
    116 051410  001410                            BEQ     340$            ;YES !!
    117 051412  012737  001000  001140  335$:     MOV     #AOE,$GDDAT     ;SET FOR AOE = 1
    118 051420  005037  051710                    CLR     505$            ;CLEAR EXPECTED TRACK
    119 051424  012737  000001  051712            MOV     #1,510$         ;EXPECT SECTOR = 1
    120 051432  013737  001350  001142  340$:     MOV     RMER1I,$BDDAT   ;BAD DATA FOR TYPEOUT
    121 051440  042737  176777  001142            BIC     #^CAOE,$BDDAT
    122 051446  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;IS AOE CORRECTY??
    123 051454  001413                            BEQ     345$            ;YES!!
    124 051456  062716  000004                    ADD     #4,(SP)
    125 051462  112776  000020  000000            MOVB    #20,@(SP)       ;ERROR NUMBER
    126 051470  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    127 051474  004736                            JSR     PC,@(SP)+       ;REPORT AOE IS WRONG
    128 051476  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
    129 051502  000240                            NOP
    130
    131                                 ;REPORT ERROR IF RMDA IS NOT CORRECT
    132 051504  032737  002000  001350  345$:     BIT     #IAE,RMER1I     ;WAS THERE AN IAE ERROR ??
    133 051512  001062                            BNE     355$            ;YES - DONT CHECK RMDA,RMDC
    134 051514  013737  051710  001140            MOV     505$,$GDDAT     ;SETUP EXPECTED DISK ADDRESS
    135 051522  000337  001140                    SWAB    $GDDAT
    136 051526  113737  051712  001140            MOVB    510$,$GDDAT
    137 051534  013737  001342  001142            MOV     RMDAI,$BDDAT    ;SETUP RECEIVED DISK ADDRESS
    138 051542  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
    139 051550  001413                            BEQ     350$            ;BRANCH IF EQUAL
    140 051552  062716  000004                    ADD     #4,(SP)
    141 051556  112776  000021  000000            MOVB    #21,@(SP)       ;ERROR NUMBER
    142 051564  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    143 051570  004736                            JSR     PC,@(SP)+       ;REPORT BAD DISK ADDRESS
    144 051572  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
    145 051576  000240                            NOP
    146
    147                                 ;REPORT ERROR IF RMDC IS INCORRECT
    148 051600  013737  051706  001140  350$:     MOV     500$,$GDDAT     ;SETUP EXPECTED CYLINDER
    149 051606  042737  176000  001140            BIC     #^C1777,$GDDAT
    150 051614  013737  001370  001142            MOV     RMDCI,$BDDAT    ;SETUP RECEIVED CYLINDER
    151 051622  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;COMPARE CYLINDERS
    152 051630  001413                            BEQ     355$            ;BRANCH IF EQUAL
    153 051632  062716  000004                    ADD     #4,(SP)
    154 051636  112776  000022  000000            MOVB    #22,@(SP)       ;ERROR NUMBER
    155 051644  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    156 051650  004736                            JSR     PC,@(SP)+       ;REPORT BAD CYLINDER
    157 051652  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
    158 051656  000240                            NOP
    159
    160 051660  062716  000004          355$:     ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
    161 051664  105776  000000                    TSTB    @(SP)           ;WAS ERROR FOUND??
    162 051670  001403                            BEQ     360$
    163 051672  062716  000004                    ADD     #4,(SP)         ;MOVE (SP) TO ERROR RETURN
    164 051676  000402                            BR      365$
    165 051700  162716  000004          360$:     SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    166 051704  000207                  365$:     RTS     PC
    167
    168 051706  000000                  500$:     .WORD   0               ;CYLINDER
    169 051710  000000                  505$:     .WORD   0               ;TRACK
    170 051712  000000                  510$:     .WORD   0               ;SECTOR
    171 051714  000000                  515$:     .WORD   0               ;FUNCTION CODE
```

```
    172 051716  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

051720                          CMPERRSTS:

                                ;MASK AND STORE THE CONTENTS OF RMER1  AND RMER2
051720  013737  001350  001176        MOV     RMER1I,$TMP1    ;STORE RMER1 AT TEMP STORAGE
051726  047637  000000  001176        BIC     @(SP),$TMP1     ;MASK   RMER1
051734  062716  000002                ADD     #2,(SP)         ;MOVE SP TO NEXT MASK
051740  013737  001376  001200        MOV     RMER2I,$TMP2    ;STORE RMER2 AT TEMP STORAGE
051746  047637  000000  001200        BIC     @(SP),$TMP2     ;MASK RMER2


                                ;CLEAR USER'S ERROR CALL
051754  062716  000006                ADD     #6,(SP)         ;MOVE SP TO USER'S ERROR CALL
051760  105076  000000                CLRB    @(SP)           ;CLEAR ERROR NUMBER
051764  162716  000004                SUB     #4,(SP)         ;LEAVE SP AT NO ERROR RETURN

                                ;SEE IF THERE WERE ANY ERRORS IN RMER1, I.E., $TMP1
051770  005737  001176                TST     $TMP1           ;ANY ERRORS TO REPORT??
051774  001420                        BEQ     5$              ;NO !!
051776  013737  001176  001142        MOV     $TMP1,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
052004  005037  001140                CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
052010  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052014  112776  000066  000000        MOVB    #66,@(SP)       ;CORRECTABLE DATA CHECK ERROR #
052022  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052026  004736                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052030  162716  000010                SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
052034  000240                        NOP
052036                          5$:


                                ;SEE IF THERE ARE ANY ERRORS TO REPORT IN RMER2 ($TMP2)
052036  005737  001200                TST     $TMP2           ;ANY ERRORS IN RMER2?
052042  001420                        BEQ     10$             ;NO!!

052044  013737  001200  001142        MOV     $TMP2,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
052052  005037  001140                CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
052056  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052062  112776  000067  000000        MOVB    #67,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
```

```
     58 052070  162716  000002                  SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
     59 052074  004736                          JSR     PC,@(SP)+        ;REPORT ERROR VIA USER
     60 052076  162716  000010                  SUB     #10,(SP)         ;MOVE SP TO NO ERROR RETURN
     61 052102  000240                          NOP
     62 052104                          10$:
     63
     64                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
     65 052104  062716  000004                  ADD     #4,(SP)          ;MOVE SP TO USER'S ERROR CALL
     66 052110  105776  000000                  TSTB    @(SP)            ;WAS THERE AN ERROR CALLED??
     67 052114  001403                          BEQ     20$              ;NO!!
     68 052116  062716  000004                  ADD     #4,(SP)          ;YES - MOVE SP TO ERROR RETURN
     69 052122  000402                          BR      30$
     70 052124  162716  000004          20$:    SUB     #4,(SP)          ;MOVE SP TO NO ERROR RETURN
     71 052130  000207                  30$:    RTS     PC               ;RETURN TO USER
     72
```

```
                        .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

052132                  DEVSEL:

                        ;CLEAR USER'S ERROR CALL
052132  062716  000004          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
052136  105076  000000          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF CALL
052142  162716  000004          SUB     #4,(SP)         ;MOVE SP BACK

                        ;SAVE USER'S INFORMATION AND SETUP REGISTERS
052146  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
052152  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
052156  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
052160  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052162  012737  052302  000004  MOV     #20$,ERRVEC     ;SETUP FOR BUS TIMEOUT
052170  012737  000300  000006  MOV     #PR6,ERRVEC+2
052176  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
052202  013701  001464          MOV     TSTQUE,R1       ;R1 POINTS TO DEVICE NUMBER

                        ;SELECT DEVICE AND VERIFY THAT DEVICE IS AVAILABLE
052206  111160  000010          MOVB    (R1),RMCS2(R0)  ;WRITE UNIT SELECT BITS
052212  016037  000000  001176  MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
052220  016037  000010  001174  MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS

052226  032737  010000  001174  BIT     #NED,$TMP0      ;IS DEVICE NONEXISTENT ?
052234  001407                  BEQ     10$             ;NO!!
052236  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
052244  112776  000111  000010  MOVB    #111,@10(SP)    ;WRITE ERROR NUMBER
052252  000422                  BR      30$

052254  032737  004000  001176  10$:    BIT     #DVA,$TMP1      ;IS DEVICE AVAILABLE ?
052262  001021                  BNE     35$             ;YES!!
052264  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
052272  112776  000112  000010  MOVB    #112,@10(SP)    ;WRITE ERROR NUMBER
052300  000407                  BR      30$

                        ;HANDLE BUS TIMEOUT
052302  022626          20$:    CMP     (SP)+,(SP)+     ;ADJUST SP
052304  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
052312  112776  000113  000010  MOVB    #113,@10(SP)    ;WRITE BUS TIMEOUT ERROR NUMBER
052320  162766  000002  000010  30$:    SUB     #2,10(SP)       ;ADJUST RETURN TO 'NOP' PRECEDING
                                                        ;THE ERROR CALL

                        ;RESTORE USERS DATA AND RETURN TO ADDRESS ON STACK
052326                  35$:
052326  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
052330  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
052332  012637  000006          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
```

```
        052336  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     52 052342  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
                        ;
052344                  SEKSTS:

                        ;CLEAR USERS' ERROR CALL
                                NOP
052344  000240
052346  062716  000004          ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
052352  105076  000000          CLRB    @(SP)           ;CLEAR ERROR NUMBER
052356  162716  000004          SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
052362  005037  053602          CLR     300$            ;CLEAR ERROR FLAGS

                        ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN WRITING
                        ;LOCAL REGISTERS, I.E., "PAR"  = 1 AND "DPE" = 0
052366  032737  000010  001350          BIT     #PAR,RMER1I     ;WAS PARITY ERROR DETECTED??
052374  001424                        BEQ     1$              ;NO!!
052376  032737  000010  001376          BIT     #DPE,RMER2I     ;WAS IT DUE TO CONTROL BUS??
052404  001020                        BNE     1$              ;NOT SURE!!

                        ;REPORT REGISTER PARITY ERROR VIA USER'S ERROR CALL
052406  005037  001140          CLR     $GDDAT          ;EXPECTED STATUS
052412  013737  001350  001142  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
052420  062716  000004          ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
052424  112776  000050  000000  MOVB    #50,@(SP)       ;ERROR #50
052432  162716  000002          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052436  004736                  JSR     PC,@(SP)+
052440  162716  000010          SUB     #10,(SP)        ;RESTORE STACK
052444  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.
052446  012737  002000  001140  1$:     MOV     #IAE,$GDDAT     ;SETUP FOR IAE = 1
052454  052737  040000  053602          BIS     #SKI,300$       ;SETUP FOR SKI = 1
052462  023727  001444  001466          CMP     RMDC0,#822.     ;GREATER THAN LAST CYLINDER ?
052470  101025                        BHI     3$              ;YES - CYLINDER IS INVALID
052472  042737  040000  053602          BIC     #SKI,300$       ;CLEAR SKI ERROR FLAG

052500  123737  001417  001333          CMPB    RMDA0+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
052506  101016                        BHI     3$              ;YES - TRACK IS INVALID

052510  123727  001416  000035          CMPB    RMDA0,#29.      ;SECTOR > 29. ?
052516  101410                        BLOS    2$              ;BR IF NO
```

```
     58 052520  032737  010000  001442          BIT     #FMT16,RMOFO    ;18 BIT FORMAT ?
     59 052526  001406                          BEQ     3$              ;YES - SECTOR IS INVALID FOR 18 BIT MODE
     60 052530  123727  001416  000037          CMPB    RMDAO,#31.      ;SECTOR > 31. ?
     61 052536  101002                          BHI     3$              ;YES - SECTOR IS INVALID
     62
     63 052540  005037  001140          2$:     CLR     $GDDAT          ;''IAE'' SHOULD = 0
     64
     65                                 ;COMPARE EXPECTED AND RECIEVED ''IAE'' STATUS
     66 052544  013737  001350  001142  3$:     MOV     RMER1I,$BDDAT   ;IS IAE OK??
     67 052552  042737  175777  001142          BIC     #^CIAE,$BDDAT   ;SAVE IAE BIT FOR COMPARE
     68 052560  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;CORRECT ''IAE'' STATUS ?
     69 052566  001004                          BNE     35$             ;BR IF NO
     70 052570  042737  040000  053602          BIC     #SKI,300$       ;CLEAR SKI FLAG
     71 052576  000413                          BR      5$              ;GO CHECK NEXT ERROR
     72 052600                          35$:
     73                                 ;REPORT INCORRECT ''IAE'' STATUS VIA USER'S ERROR CALL
     74 052600  062716  000004                  ADD     #4,(SP)
     75 052604  112776  000051  000000          MOVB    #51,@(SP)       ;ERROR 51
     76 052612  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     77 052616  004736                          JSR     PC,@(SP)+       ;REPORT INCORRECT IAE
     78 052620  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP)
     79 052624  000240                          NOP
     80 052626                          5$:
     81
     82                                 ;REPORT ANY IVC ERROR AS
     83                                 ;       IVC ERROR WITH VOLUME VALID ZERO
     84                                 ;       ERRONEOUS IVC ERROR, VOLUME VALID IS SET
     85 052626  032737  010000  001376          BIT     #IVC,RMER2I     ;IVC ERROR??
     86 052634  001427                          BEQ     52$             ;NO!!
     87 052636  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
     88 052642  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     89 052650  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
     90 052654  112776  000060  000000          MOVB    #60,@(SP)       ;ERROR 60 IF VV = 0
     91 052662  032737  000100  001346          BIT     #VV,RMDSI
     92 052670  001403                          BEQ     51$
     93 052672  112776  000061  000000          MOVB    #61,@(SP)       ;ERROR 61 IF VV  = 1
     94 052700  162716  000002          51$:    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     95 052704  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     96 052706  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
     97 052712  000240                          NOP
     98
     99 052714  013737  001376  001142  52$:    MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
    100 052722  042737  137777  001142          BIC     #^CSKI,$BDDAT   ;CLEAR DONT CARES
    101 052730  013737  053602  001140          MOV     300$,$GDDAT     ;GET EXPECTED SKI STATUS
    102 052736  042737  137777  001140          BIC     #^CSKI,$GDDAT   ;CLEAR DONT CARES
    103 052744  001417                          BEQ     53$             ;BRANCH IF 0 EXPECTED
    104
    105                                 ;REPORT ERROR IF SKI IS NOT SET (IAE WAS NOT DETECTED)
    106 052746  032737  040000  001142          BIT     #SKI,$BDDAT     ;WAS SKI DETECTED ??
    107 052754  001032                          BNE     54$             ;YES !!
    108 052756  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    109 052762  112776  000267  000000          MOVB    #267,@(SP)      ;WRITE ERROR NUMBER
    110 052770  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    111 052774  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    112 052776  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    113 053002  000443                          BR      6$              ;GO TO NEXT ERROR CHECK
    114 053004                          53$:
```

```
                                        ;REPORT ERROR IF SKI IS SET
053004  032737  040000  001142          BIT     #SKI,$BDDAT     ;IS SKI SET ??
053012  001413                          BEQ     54$             ;NO - SKI IS OK
053014  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
053020  112776  000054  000000          MOVB    #54,@(SP)       ;LOAD ERROR NUMBER
053026  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053032  004736                          JSR     PC,@(SP)+       ;REPORT SEEK ERROR
053034  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
053040  000240                          NOP

                                        ;REPORT ANY DEVICE CHECK
053042  032737  000200  001376  54$:    BIT     #DVC,RMER2I     ;WAS THERE DVC DURING SEEK??
053050  001420                          BEQ     6$              ;NO!!
053052  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
053056  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
053064  062716  000004                  ADD     #4,(SP)
053070  112776  000055  000000          MOVB    #55,@(SP)       ;ERROR  #55
053076  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053102  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053104  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
053110  000240                          NOP

                                        ;REPORT ANY 'OPI' ERROR AS OPI WITH MOL  = 0, OR  OPI
                                        ;BECAUSE ON CYLINDER LATCH DIDN'T RESET
053112  032737  020000  001350  6$:     BIT     #OPI,RMER1I     ;'OPI' ERROR??
053120  001427                          BEQ     8$              ;NO!!
053122  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
053126  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053134  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
053140  112776  000052  000000          MOVB    #52,@(SP)       ;LOAD ERROR NUMBER
053146  032737  010000  001346          BIT     #MOL,RMDSI      ;WAS MEDIUM ON LINE??
053154  001403                          BEQ     7$              ;NO!!
053156  112776  000053  000000          MOVB    #53,@(SP)       ;YES - CHANGE ERROR NUMBER
053164  162716  000002          7$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053170  004736                          JSR     PC,@(SP)+       ;REPORT 'OPI' ERROR
053172  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
053176  000240                          NOP

                                        ;SEE IF 'PIP' = 0, AND 'ATA', 'MOL' AND 'VV' = 1
053200  013746  001346          8$:     MOV     RMDSI,-(SP)
053204  042716  047677                  BIC     #^C<ATA!PIP!MOL!VV>,(SP)
053210  022726  110100                  CMP     #ATA!MOL!VV,(SP)+
053214  001002                          BNE     9$              ;ERROR IN RMDS
053216  000137  053552                  JMP     14$             ;RMDS IS OK

                                        ;REPORT ERROR IF MOL = 0 AND OPI = 0
053222  032737  010000  001346  9$:     BIT     #MOL,RMDSI      ;IS MOL RESET??
053230  001030                          BNE     10$             ;NO - MOL IS SET
053232  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET
053240  001024                          BNE     10$             ;YES - DONT REPORT ERROR
053242  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053250  052737  010000  001140          BIS     #MOL,$GDDAT
053256  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053264  062716  000004                  ADD     #4,(SP)
053270  112776  000062  000000          MOVB    #62,@(SP)
053276  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
```

```
                                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                        SUB     #10,(SP)
                                        NOP

                                ;REPORT AN ERROR IF 'PIP' IS STIL SET AND SKI NOT SET
                        10$:    BIT     #PIP,RMDSI      ;IS 'PIP' STILL SET??
                                BEQ     11$             ;NO!!
                                BIT     #SKI,RMER2I     ;WAS 'SKI' SET??
                                BNE     11$             ;YES-DONT REPORT PIP
                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                BIC     #PIP,$BDDAT
                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                ADD     #4,(SP)         ;MOVE (SP) TO ERROR
                                MOVB    #56,@(SP)       ;LOAD ERROR NUMBER
                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                JSR     PC,@(SP)+       ;REPORT 'PIP' SET AFTER SEEK
                                SUB     #10,(SP)                ;RESTORE (SP)
                                NOP

                                ;REPORT AN ERROR IF "ATA" IS NOT SET
                        11$:    BIT     #ATA,RMDSI      ;WAS "ATA" SET ??
                                BNE     13$             ;YES!!
                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                BIS     #ATA!MOL!DPR!DRY,$GDDAT
                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                ADD     #4,(SP)         ;MOVE (SP) TO ERROR
                                MOVB    #57,@(SP)       ;LOAD ERROR NUMBER
                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                JSR     PC,@(SP)+       ;REPORT ATTENTION NOT SET DURING
                                                        ;SEEK TEST
                                SUB     #10,(SP)                ;RESTORE (SP)
                                NOP

                                ;REPORT ERROR IF VOLUME VALID IS RESET AND IVC IS ZERO
                        13$:    BIT     #VV,RMDSI       ;IS VV = 0 ??
                                BNE     14$             ;NO!!
                                BIT     #IVC,RMER2I     ;IS IVC ALSO 0 ??
                                BNE     14$             ;NO - IVC IS SET
                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                BIS     #VV,$GDDAT
                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                ADD     #4,(SP)
                                MOVB    #64,@(SP)       ;ERROR #64
                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                JSR     PC,@(SP)+
                                SUB     #10,(SP)
                                NOP
                        14$:

                                ;MODIFY THE RETURN ADDRESS IF AN ERROR WAS DETECTED
                                NOP
                                ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
                                TSTB    @(SP)           ;WAS ERROR CALLED??
                                BEQ     15$             ;NO!!
                                ADD     #4,(SP)         ;MOVE TO ERROR RETURN
                                BR      16$
```

```
    229 053574  162716  000004          15$:    SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    230 053600  000207                  16$:    RTS     PC              ;RETURN
    231
    232 053602  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
                        ;       ???

053604                  CNTCLR:
053604  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
053606  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
053610  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
053614  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
053620  012737  053660  000004  MOV     #10$,ERRVEC     ;SETUP FOR BUS TIMEOUT
053626  012737  000300  000006  MOV     #PR6,ERRVEC+2
053634  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
053640  012760  000040  000010  MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
053646  013701  001464          MOV     TSTQUE,R1       ;GET DEVICE UNDER TEST
053652  111160  000010          MOVB    (R1),RMCS2(R0)  ;SELECT DEVICE
053656  000412                  BR      20$

053660  022626          10$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
053662  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
053670  112776  000007  000010  MOVB    #7,@10(SP)      ;WRITE THE ERROR NUMBER
053676  162766  000002  000010  SUB     #2,10(SP)       ;ADJUST SP TO RETURN TO ERROR
053704                  20$:
053704  012637  000006          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
053710  012637  000004          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
053714  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
053716  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
053720  000207                  RTS     PC
```

```
                                .SBTTL  CONTROLLER CLEAR STATUS CHECK SUBROUTINE

                                ;THIS SUBROUTINE VERIFIES THAT THE SUBSYSTEM IS INITIALIZED BASED ON
                                ;STATUS STORED IN THE GET BUFFER.  THIS SUBROUTINE SHOULD ONLY BE
                                ;USED FOLLOWING A CONTROLLER CLEAR OPERATION, I.E., WRITING A 1 IN BIT
                                ;5 OF RMCS2, BECAUSE THE ERROR MESSAGES ARE BASED ON THAT CONDITION.

                                ;STATUS PERTINENT TO THE DEVICE IS NOT CHECKED.  IN PARTICULAR, THE
                                ;FOLLOWING STATUS BITS ARE NOT CHECKED:
                                ;
                                ;       ATA,ERR,PIP,MOL,WRL,LBT,PGM,VV,OM,UNS,SKI,DVC
                                ;

                                ;CALL:
                                ;(1)    JSR     PC,CLRSTS
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS

053722                          CLRSTS:

                                ;CLEAR USER'S ERROR CALL
053722  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR
053726  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
053732  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR
                                ;REPORT ERROR IF RMCS1 NOT INITIALIZED
053736  013737  001334  001142  4$:     MOV     RMCS1I,$BDDAT   ;VERIFY RMCS1
053744  042737  100000  001142          BIC     #SC,$BDDAT      ;IGNORE SPECIAL CONDITION
053752  012737  004200  001140          MOV     #DVA!RDY,$GDDAT ;EXPECT DVA & RDY
053760  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED, RECEIVED
053766  001413                          BEQ     5$              ;BRANCH IF EQUAL
053770  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053774  112776  000126  000000          MOVB    #126,@(SP)      ;WITE ERROR NUMBER IN CALL
054002  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054006  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
054010  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
054014  000240                          NOP
                                ;REPORT ERROR IF RMBA NOT RESET
054016  005037  001140          5$:     CLR     $GDDAT          ;VERIFY RMBA IS ZERO
054022  013737  001340  001142          MOV     RMBAI,$BDDAT
054030  001413                          BEQ     7$              ;BRANCH IF ZERO
054032  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054036  112776  000127  000000          MOVB    #127,@(SP)      ;WITE ERROR NUMBER IN CALL
054044  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054050  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
054052  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
054056  000240                          NOP
                                ;REPORT ERROR IF RMCS2 NOT INITIALIZED
054060  013737  001344  001142  7$:     MOV     RMCS2I,$BDDAT   ;VERIFY RMCS2
054066  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
054070  005046                          CLR     -(SP)           ;EXPECT IR & UNIT NUMBER
054072  013701  001464                  MOV     TSTQUE,R1       ;R1 = ADDRESS OF TEST QUE
054076  111116                          MOVB    (R1),(SP)
054100  052716  000100                  BIS     #IR,(SP)
054104  012637  001140                  MOV     (SP)+,$GDDAT
```

```
     058 054110  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
     059 054112  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
     060 054120  001413                          BEQ     9$              ;BRANCH IF EQUAL
     061 054122  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     062 054126  112776  000130  000000          MOVB    #130,@(SP)      ;WITE ERROR NUMBER IN CALL
     063 054134  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     064 054140  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     065 054142  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     066 054146  000240                          NOP
     067                             ;REPORT ERROR IF RMER1 NOT RESET-IGNORE UNS
     068 054150  005037  001140      9$:         CLR     $GDDAT          ;VERIFY RMER1
     069 054154  013737  001350  001142          MOV     RMER1I,$BDDAT
     070 054162  042737  040000  001142          BIC     #UNS,$BDDAT     ;IGNORE UNSAFE
     071 054170  001413                          BEQ     13$             ;BRANCH IF ZERO
     072 054172  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     073 054176  112776  000131  000000          MOVB    #131,@(SP)      ;WITE ERROR NUMBER IN CALL
     074 054204  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     075 054210  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     076 054212  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     077 054216  000240                          NOP
     078                             ;REPORT ERROR IF RMMR1 NOT INITIALIZED-IGNORE WC,LS,LST
     079 054220  013737  001360  001142  13$:     MOV     RMMR1I,$BDDAT   ;VERIFY RMMR
     080 054226  042737  000046  001142          BIC     #WC!LS!LST,$BDDAT       ;IGNORE WORD CLOCK, SCT, TRK
     081 054234  012737  000010  001140          MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA BIT
     082 054242  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
     083 054250  001413                          BEQ     17$             ;BRANCH IF 0
     084 054252  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     085 054256  112776  000133  000000          MOVB    #133,@(SP)      ;WITE ERROR NUMBER IN CALL
     086 054264  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     087 054270  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     088 054272  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     089 054276  000240                          NOP
     090                             ;REPORT AN ERROR IF RMEC2 IS NOT RESET
     091 054300  005037  001140      17$:        CLR     $GDDAT          ;EXPECT ZEROS
     092 054304  013737  001402  001142          MOV     RMEC2I,$BDDAT   ;VERIFY RMEC2=0
     093 054312  001413                          BEQ     19$
     094 054314  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     095 054320  112776  000135  000000          MOVB    #135,@(SP)      ;WITE ERROR NUMBER IN CALL
     096 054326  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     097 054332  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     098 054334  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     099 054340  000240                          NOP
     100                             ;REPORT ERROR IF RMMR2 NOT INITIALIZED-IGNORE RQA,RQB
     101 054342  013737  001374  001142  19$:     MOV     RMMR2I,$BDDAT   ;VERIFY RMMR2
     102 054350  042737  140000  001142          BIC     #RQA!RQB,$BDDAT
     103 054356  012737  011777  001140          MOV     #TST!1777,$GDDAT        ,EXPECT TEST BIT ON
     104 054364  023737  001140  001142          CMP     $GDDAT,$BDDAT
     105 054372  001413                          BEQ     21$
     106 054374  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     107 054400  112776  000136  000000          MOVB    #136,@(SP)      ;WITE ERROR NUMBER IN CALL
     108 054406  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     109 054412  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     110 054414  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     111 054420  000240                          NOP
     112                             ;REPORT ERROR IF RMER2 NOT RESET-IGNORE SKI,DVC
     113 054422  005037  001140      21$:        CLR     $GDDAT          ;EXPECT ALL ZEROS
     114 054426  013737  001376  001142          MOV     RMER2I,$BDDAT   ;VERIFY RMER2
```

```
    115 054434  042737  040200  001142          BIC     #SKI!DVC,$BDDAT ;IGNORE DEVICE ERRORS
    116 054442  001413                          BEQ     215$            ;BRANCH IF OTHER BITS 0
    117 054444  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    118 054450  112776  000174  000000          MOVB    #174,@(SP)      ;WITE ERROR NUMBER IN CALL
    119 054456  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    120 054462  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    121 054464  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    122 054470  000240                          NOP
    123                             ;REPORT ERROR IF RMDS NOT INITIALIZED
    124 054472  013737  001346  001142  215$:   MOV     RMDSI,$BDDAT    ;TEST DRIVE STATUS REGISTER
    125 054500  042737  177777  001142          BIC     #^C<DRY!DPR>,$BDDAT
    126 054506  012737  000600  001140          MOV     #DPR!DRY,$GDDAT ;EXPECTED DRIVE STATUS
    127 054514  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
    128 054522  001413                          BEQ     22$             ;BRANCH IF EQUAL
    129 054524  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    130 054530  112776  000134  000000          MOVB    #134,@(SP)      ;WRITE ERROR NUMBER
    131 054536  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    132 054542  004736                          JSR     PC,@(SP)+       ;REPORT ERROR TO USER
    133 054544  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    134 054550  000240                          NOP
    135 054552  062716  000004          22$:    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    136 054556  105776  000000                  TSTB    @(SP)           ;WAS AN ERROE DETECTED??
    137 054562  001403                          BEQ     23$             ;NO!!
    138 054564  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
    139 054570  000402                          BR      24$
    140 054572  162716  000004          23$:    SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
    141 054576  000240                  24$:    NOP
    142 054600  000207                          RTS     PC
    143
```

```
                        .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

054602                  ACKSTS:

                        ;CLEAR USER'S ERROR CALL
054602  062716  000004          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054606  105076  000000          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE
054612  162716  000004          SUB     #4,(SP)         ;MOVE SP BACK

                        ;REPORT AN ERROR  IF 'VV' IS 0
054616  032737  000100  001346  BIT     #VV,RMDSI       ;IS VOLUME VALID SET??
054624  001024                  BNE     1$              ;YES!!
054626  013737  001346  001140  MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
054634  052737  000100  001140  BIS     #VV,$GDDAT
054642  013737  001346  001142  MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
054650  062716  000004          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054654  112776  000155  000000  MOVB    #155,@(SP)      ;WRITE NUMBER IN ERROR CALL
054662  162716  000002          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054666  004736                  JSR     PC,@(SP)+       ;REPORT THE ERROR
054670  162716  000010          SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
054674  000240                  NOP
054676                  1$:

                        ;REPORT AN ERROR IF 'MOL' IS 0
054676  032737  010000  001346  BIT     #MOL,RMDSI      ;IS MOL SET??
054704  001024                  BNE     2$              ;YES!!
054706  013737  001346  001140  MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
054714  052737  010000  001140  BIS     #MOL,$GDDAT
054722  013737  001346  001142  MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
054730  062716  000004          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054734  112776  000041  000000  MOVB    #41,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
054742  162716  000002          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054746  004736                  JSR     PC,@(SP)+       ;REPORT TH ERROR
054750  162716  000010          SUB     #10,(SP)        ;MOVE SP TO BRANCH
054754  000240                  NOP
054756                  2$:

                        ;SEE IF 'UNS','OPI','RMR','ILR', OR 'ILF' IS SET
054756  032737  060007  001350  BIT     #UNS!OPI!RMR!ILR ILF,RMER1I
054764  001570                  BEQ     7$

                        ;REPORT AN ERROR IF 'UNS' IS SET
054766  032737  040000  001350  BIT     #UNS,RMER1I     ;WAS UNS SET??
054774  001424                  BEQ     3$              ;NO!.
054776  013737  001350  001142  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

```
     58 055004  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
     59 055012  042737  040000  001140          BIC     #UNS,$GDDAT
     60 055020  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
     61 055024  112776  000042  000000          MOVB    #42,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
     62 055032  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     63 055036  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
     64 055040  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     65 055044  000240                          NOP
     66 055046                          3$:
     67
     68                                 ;REPORT ANY OPI ERROR
     69 055046  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET ??
     70 055054  001424                          BEQ     4$              ;NO!!
     71 055056  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
     72 055064  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
     73 055072  042737  020000  001140          BIC     #OPI,$GDDAT
     74 055100  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
     75 055104  112776  000043  000000          MOVB    #43,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
     76 055112  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     77 055116  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
     78 055120  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     79 055124  000240                          NOP
     80 055126                          4$:
     81
     82                                 ;REPORT ANY RMR ERROR
     83 055126  032737  000004  001350          BIT     #RMR,RMER1I     ;WAS RMR SET??
     84 055134  001424                          BEQ     5$              ;NO!!
     85 055136  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
     86 055144  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
     87 055152  042737  000004  001140          BIC     #RMR,$GDDAT
     88 055160  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
     89 055164  112776  000044  000000          MOVB    #44,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
     90 055172  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     91 055176  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
     92 055200  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     93 055204  000240                          NOP
     94 055206                          5$:
     95
     96                                 ;REPORT ANY     ILR ERROR
     97 055206  032737  000002  001350          BIT     #ILR,RMER1I     ;WAS ILR SET??
     98 055214  001424                          BEQ     6$              ;NO!!
     99 055216  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    100 055224  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
    101 055232  042737  000002  001140          BIC     #ILR,$GDDAT
    102 055240  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    103 055244  112776  000045  000000          MOVB    #45,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
    104 055252  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    105 055256  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
    106 055260  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
    107 055264  000240                          NOP
    108 055266                          6$:
    109
    110                                 ;REPORT ANY     ILF ERROR
    111 055266  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS ILF SET??
    112 055274  001424                          BEQ     7$              ;NO!!
    113 055276  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    114 055304  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
```

```
    115 055312  042737  000001  001140          BIC     #ILF,$GDDAT
    116 055320  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    117 055324  112776  000046  000000          MOVB    #46,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
    118 055332  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    119 055336  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
    120 055340  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
    121 055344  000240                          NOP
    122 055346                          7$:
    123
    124                                 ;AUGMENT RETURN ADDRESS IF ERROR WAS FOUND
    125 055346  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    126 055352  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
    127 055356  001403                          BEQ     8$              ;NO!!
    128 055360  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
    129 055364  000402                          BR      9$
    130 055366  162716  000004          8$:     SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
    131 055372  000240                  9$:     NOP
    132 055374  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

055376                          RCLSTS:

                                ;CLEAR USER'S ERROR NUMBER
055376  062716  000004                  ADD     #4,(SP)
055402  105076  000000                  CLRB    @(SP)           ;CLEAR USER'S ERROR CALL
055406  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO BRANCH


                                ;SEE IF 'PAR' OR 'ILF' OR 'OPI' OR 'IAE' IS SET
055412  032737  022011  001350          BIT     #OPI!PAR!ILF!IAE,RMER1I
055420  001553                  BEQ     4$              ;NONE ARE SET - GO TO NEXT CHECK

                                ;REPORT ANY MASSBUS CONTROL BUS PARITY ERROR, I.E.,
                                ;'PAR' = 1 AND 'DPE' = 0
055422  032737  000010  001350          BIT     #PAR,RMER1I     ;WAS 'PAR' SET??
055430  001430                  BEQ     1$              ;NO!!
055432  032737  000010  001376          BIT     #DPE,RMER2I     ;WAS 'DPE' SET??
055440  001024                  BNE     1$              ;YES - NOT A REGISTER ERROR
055442  013737  001350  001140          MOV RMER1I,$GDDAT       ;EXPECTED STATUS
055450  042737  000010  001140          BIC     #PAR,$GDDAT
055456  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055464  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
055470  112776  000050  000000          MOVB    #50,@(SP)       ;WRITE ERROR NUMBER IN CALL
055476  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055502  004736                  JSR     PC,@(SP)+       ;GO REPORT ERROR
055504  162716  00C010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
055510  000240                  NOP
055512                          1$:

                                ;REPORT ANY 'ILF' ERROR
055512  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS 'ILF' SET??
055520  001424                  BEQ     2$              ;NO!!
055522  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055530  042737  000001  001140          BIC     #ILF,$GDDAT
055536  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055544  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055550  112776  000071  000000          MOVB    #71,@(SP)       ;WRITE ERROR NUMBER IN CALL
055556  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055562  004736                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055564  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
055570  000240                  NOP
055572                          2$:
```

```
                                        ;REPORT ANY 'OPI' ERROR AS
                                        ;       . OPI DUE TO 'MOL' = 0
                                        ;       . OPI BECAUSE ON CYLINDER LATCH DIDN'T RESET
055572  032737  020000  001350            BIT     #OPI,RMER1I     ;WAS OPI SET??
055600  001433                            BEQ     31$             ;NO!!
055602  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055610  042737  020000  001140            BIC     #OPI,$GDDAT
055616  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055624  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055630  112776  000072  000000            MOVB    #72,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
055636  032737  010000  001346            BIT     #MOL,RMDSI      ;WAS 'MOL' = 0??
055644  001403                            BEQ     3$              ;YES!!
055646  112776  000073  000000            MOVB    #73,@(SP)       ;NO - CHANGE ERROR NUMBER
055654  162716  000002            3$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055660  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055662  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
055666  000240                            NOP
055670                            31$:

                                        ;REPORT AN ERROR IF "IAE" IS SET
055670  032737  002000  001350            BIT     #IAE,RMER1I     ;IS "IAE" SET??
055676  001424                            BEQ     4$              ;NO!!
055700  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055706  042737  002000  001140            BIC     #IAE,$GDDAT
055714  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055722  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
055726  112776  000070  000000            MOVB    #70,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
055734  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055740  004736                            JSR     PC,@(SP)+       ;REPORT ERROR
055742  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
055746  000240                            NOP
055750                            4$:

                                        ;SEE IF "SKI" OR "IVC" OR "DVC" IS SET
055750  032737  050200  001376            BIT     #SKI!IVC!DVC,RMER2I
055756  001517                            BEQ     8$              ;NONE OF THE BITS ARE SET


                                        ;REPORT ANY "IVC" ERROR AS
                                        ;       .IVC WITH VV = 0
                                        ;       .ERRONEOUS IVC ERROR
055760  032737  010000  001376            BIT     #IVC,RMER2I     ;WAS IVC SET??
055766  001433                            BEQ     6$              ;NO!!
055770  013737  001376  001140            MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
055776  042737  010000  001140            BIC     #IVC,$GDDAT
056004  013737  001376  001142            MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
056012  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056016  112776  000074  000000            MOVB    #74,@(SP)       ;WRITE ERROR NUMBER IN CALL
056024  032737  000100  001346            BIT     #VV,RMDSI       ;WAS VV = 0??
056032  001403                            BEQ     5$              ;YES!!
056034  112776  000075  000000            MOVB    #75,@(SP)       ;NO - CHANGE ERROR NUMBER
056042  162716  000002            5$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056046  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056050  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
056054  000240                            NOP
056056                            6$:
```

```
                                        ;REPORT ANY 'SKI' ERROR
056056  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS SKI SET??
056064  001424                          BEQ     7$              ;NO!!
056066  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
056074  042737  040000  001140          BIC     #SKI,$GDDAT
056102  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
056110  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056114  112776  000076  000000          MOVB    #76,@(SP)       ;WRITE ERROR NUMBER
056122  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056126  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056130  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO BRANCH
056134  000240                          NOP
056136                          7$:

                                        ;REPORT ANY 'DVC' ERROR
056136  032737  000200  001376          BIT     #DVC,RMER2I     ;WAS 'DVC' SET??
056144  001424                          BEQ     8$              ;NO!!
056146  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
056154  042737  000200  001140          BIC     #DVC,$GDDAT
056162  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
056170  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056174  112776  000077  000000          MOVB    #77,@(SP)       ;WRITE ERROR NUMBER
056202  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056206  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056210  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
056214  000240                          NOP
056216                          8$:

                                        ;SEE IF 'PIP' AND 'OM' ARE 0, AND 'ATA','MOL' AND 'VV' ARE 1
056216  013746  001346                  MOV     RMDSI,-(SP)     ;PUT RMDS ON STACK
056222  042716  047676                  BIC     #^C<PIP!MOL!VV!OM!ATA>,(SP)
056226  022726  110100                  CMP     #ATA!MOL!VV,(SP)+
056232  001002                          BNE     85$
056234  000137  056650                  JMP     13$
056240                          85$:

                                        ;REPORT AN ERROR IF MOL = 0 AND OPI = 0, I.E., MEDIUM WENT OFF
                                        ;LINE AFTER RECALIBRATE WAS INITIATED
056240  032737  010000  001346          BIT     #MOL,RMDSI      ;DID MOL DROP??
056246  001030                          BNE     9$              ;NO!!
056250  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
056256  001024                          BNE     9$              ;YES - DON'T REPORT MOL=0
056260  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056266  052737  010000  001140          BIS     #MOL,$GDDAT
056274  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056302  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
056306  112776  000100  000000          MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
056314  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056320  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056322  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
056326  000240                          NOP
056330                          9$:

                                        ;REPORT AN ERROR IF 'VV' = 0 AND 'IVC' = 0
056330  032737  000100  001346          BIT     #VV,RMDSI       ;DID 'VV' DROP??
056336  001030                          BNE     10$             ;NO!!
056340  032737  010000  001376          BIT     #IVC,RMER2I     ;WAS THERE A IVC ERROR??
```

```
                                        BNE     10$             ;YES - DONT REPORT VV=0
056346  001024
056350  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056356  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056364  052737  000100  001140        BIS     #VV,$GDDAT
056372  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056376  112776  000101  000000        MOVB    #101,@(SP)      ;WRITE ERROR NUMBER IN CALL
056404  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056410  004736                        JSR     PC,@(SP)+
056412  162716  000010                SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
056416  000240                        NOP
056420                          10$:

                                ;REPORT AN ERROR IF ATA IS NOT SET
056420  032737  100000  001346        BIT     #ATA,RMDSI      ;WAS ATA SET DURING RECALIBRATE??
056426  001024                        BNE     11$             ;YES!!
056430  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056436  052737  100000  001140        BIS     #ATA,$GDDAT
056444  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056452  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056456  112776  000102  000000        MOVB    #102,@(SP)      ;WRITE ERROR NUMBER IN CALL
056464  162716  000002                SUB     #2,(SP)
056470  004736                        JSR     PC,@(SP)+
056472  162716  000010                SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
056476  000240                        NOP

056500                          11$:

                                ;REPORT AN ERROR IF 'OM' IS NOT ZERO BECAUSE RECALIBRATE SHOULD
                                ;ALWAYS CLEAR OFFSET MODE
056500  032737  000001  001346        BIT     #OM,RMDSI       ;WAS 'OM' RESET??
056506  001424                        BEQ     12$             ;YES!!
056510  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056516  042737  000001  001140        BIC     #OM,$GDDAT
056524  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056532  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056536  112776  000103  000000        MOVB    #103,@(SP)      ;WRITE ERROR NUMBER
056544  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056550  004736                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056552  162716  000010                SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
056556  000240                        NOP
056560                          12$:

                                ;REPORT AN ERROR IF 'PIP' IS STIL ON, I.E., DRIVE NOT ON
                                ;CYLINDER
056560  032737  020000  001346        BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
056566  001430                        BEQ     13$             ;NO!!
056570  032737  040000  001376        BIT     #SKI,RMER2I     ;WAS 'SKI' DETECTED??
056576  001024                        BNE     13$             ;YES-DONT REPORT 'PIP'
056600  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
056606  042737  020000  001140        BIC     #PIP,$GDDAT
056614  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
056622  062716  000004                ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
056626  112776  000104  000000        MOVB    #104,@(SP)      ;WRITE ERROR NUMBER
056634  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056640  004736                        JSR     PC,@(SP)+
056642  162716  000010                SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
056646  000240                        NOP
```

```
                                13$:
                                ;SEE IF ''ILR'' OR 'RMR'' OR 'UNS'' IS SET
056650  032737  040006  001350          BIT     #ILR!RMR!UNS,RMER1I
056656  001514                          BEQ     16$
                                ;REPORT AN ERROR IF ''ILR'' IS SET
056660  032737  000002  001350          BIT     #ILR,PMER1I     ;WAS ILR SET DURING RECALIBRATE??
056666  001424                          BEQ     14$             ;NO!!
056670  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056676  042737  000002  001140          BIC     #ILR,$GDDAT
056704  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056712  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
056716  112776  000105  000000          MOVB    #105,@(SP)      ;WRITE ERROR NUMBER IN CALL
056724  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056730  004736                          JSR     PC,@(SP)+
056732  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
056736  000240                          NOP
056740                          14$:
                                ;REPORT AN ERROR IF 'RMR'' IS SET
056740  032737  000004  001350          BIT     #RMR,RMER1I     ;WAS RMR SET??
056746  001424                          BEQ     15$             ;NO!!
056750  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056756  042737  000004  001140          BIC     #RMR,$GDDAT
056764  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056772  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056776  112776  000106  000000          MOVB    #106,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
057004  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057010  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
057012  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
057016  000240                          NOP
057020                          15$:
                                ;REPORT AN ERROR IF 'UNS'' IS SET AND 'DVC'' IS 0
057020  032737  040000  001350          BIT     #UNS,RMER1I     ;WAS UNSAFE ON??
057026  001430                          BEQ     16$             ;NO!!
057030  032737  000200  001376          BIT     #DVC,RMER2I     ;WAS THERE A DEVICE CHECK??
057036  001024                          BNE     16$             ;YES - DON'T REPORT UNSAFE
057040  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
057046  042737  040000  001140          BIC     #UNS,$GDDAT
057054  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
057062  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057066  112776  000107  000000          MOVB    #107,@(SP)      ;WRITE ERROR NUMBER
057074  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057100  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
057102  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
057106  000240                          NOP
057110                          16$:
                                ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
057110  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057114  105776  000000                  TSTB    @(SP)           ;WAS AN ERROR REPORTED??
057120  001403                          BEQ     17$             ;NO!!
057122  062716  000004                  ADD     #4,(SP)         ;YES - AUGMENT SP RETURN
057126  000402                          BR      18$
057130  162716  000004          17$:    SUB     #4,(SP) ;NO ERROR - RETURN SP TO BRANCH
```

```
    286 057134  000240                18$:    NOP
    287 057136  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

057140                                  DRVSTS:

                                        ;CLEAR USER'S ERROR CALL
057140  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
057144  105076  000000                          CLRB    @(SP)           ;CLEAR ERROR CALL
057150  162716  000004                          SUB     #4,(SP)         ;MOVE SP TO USER'S BRANCH
                                        ;REPORT ERROR IF RMCS1 NOT INITIALIZED
057154  013737  001334  001142         4$:      MOV     RMCS1I,$BDDAT   ;CHECK RMCS1
057162  042737  173700  001142                  BIC     #^C<DVA!FNCMSK>,$BDDAT  ;CLEAR DONT CARES
057170  012737  004010  001140                  MOV     #DVA!DRVCLR,$GDDAT      ;EXPECT DVA
057176  023737  001140  001142                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
057204  001443                                  BEQ     6$              ;BRANCH IF EQUAL
057206  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
057212  112776  000141  000000                  MOVB    #141,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
057220  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057224  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
057226  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
057232  000240                                  NOP
                                        ;REPORT ERROR IF RMDS NOT INITIALIZED
057234  013737  001346  001142         5$:      MOV     RMDSI,$BDDAT    ;CHECK RMDS
057242  042737  021101  001142                  BIC     #PGM!OM!VV!PIP,$BDDAT   ;CLEAR DONT CARES
057250  012737  010600  001140                  MOV     #MOL!DPR!DRY,$GDDAT     ;EXPECT DRY & DPR
057256  023737  001140  001142                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
057264  001413                                  BEQ     6$              ;BRANCH IF EQUAL
057266  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
057272  112776  000142  000000                  MOVB    #142,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
057300  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057304  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
057306  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
057312  000240                                  NOP
                                        ;REPORT ERROR IF RMER1 NOT INITIALIZED
057314  005037  001140                 6$:      CLR     $GDDAT          ;EXPECT 0'S
057320  013737  001350  001142                  MOV     RMER1I,$BDDAT   ;CHECK RMER1
057326  001413                                  BEQ     8$              ;BRANCH IF EQUAL
057330  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
057334  112776  000143  000000                  MOVB    #143,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
057342  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057346  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
057350  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
057354  000240                                  NOP
                                        ;REPORT ERROR IF ATA NOT INITIALIZED
057356  013737  001352  001142         8$:      MOV     RMASI,$BDDAT    ;CHECK ATTENTION BIT
057364  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
057366  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
057370  013701  001464                          MOV     TSTQUE,R1
057374  116102  000001                          MOVB    1(R1),R2
057400  042702  177400                          BIC     #^CATNMSK,R2
057404  005102                                  COM     R2
057406  040237  001142                          BIC     R2,$BDDAT
```

```
 58 057412  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 59 057414  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 60 057416  005737  001142                  TST     $BDDAT          ;IS ATTENTION CLEARED??
 61 057422  001413                          BEQ     9$              ;BRANCH IF ATTENTION CLEARED
 62 057424  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 63 057430  112776  000144  000000          MOVB    #144,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 64 057436  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 65 057442  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 66 057444  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 67 057450  000240                          NOP
 68                                 ;REPORT ERROR IF RMMR1 NOT INITIALIZED
 69 057452  013737  001360  001142  9$:     MOV     RMMR1I,$BDDAT   ;CHECK RMMR
 70 057460  042737  000046  001142          BIC     #WC!LS!LST,$BDDAT       ;CLEAR DONT CARES
 71 057466  012737  000010  001140          MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA ON
 72 057474  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
 73 057502  001413                          BEQ     11$             ;BRANCH IF ZERO
 74 057504  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 75 057510  112776  000145  000000          MOVB    #145,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 76 057516  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 77 057522  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 78 057524  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 79 057530  000240                          NOP
 80                                 ;REPORT ERROR IF RMMR2 NOT INITIALIZED
 81 057532  013737  001374  001142  11$:     MOV     RMMR2I,$BDDAT   ;CHECK RMMR2
 82 057540  042737  140000  001142          BIC     #RQA!RQB,$BDDAT ;CLEAR REQA, REQB
 83 057546  012737  011777  001140          MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
 84 057554  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 85 057562  001413                          BEQ     15$             ;BRANCH IF EQUAL
 86 057564  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 87 057570  112776  000146  000000          MOVB    #146,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 88 057576  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 89 057602  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 90 057604  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 91 057610  000240                          NOP
 92 057612  005037  001140          15$:    CLR     $GDDAT          ;EXPECT ZEROS
 93                                 ;REPORT ERROR IF RMEC2 NOT RESET
 94 057616  013737  001402  001142          MOV     RMEC2I,$BDDAT   ;CHECK RMEC2
 95 057624  001413                          BEQ     17$             ;BRANCH IF 0
 96 057626  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 97 057632  112776  000150  000000          MOVB    #150,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 98 057640  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 99 057644  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
100 057646  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
101 057652  000240                          NOP
102                                ;REPORT ERROR IF RMER2 NOT RESET
103 057654  013737  001376  001142  17$:     MOV     RMER2I,$BDDAT   ;CHECK RMER2
104 057662  001413                          BEQ     18$             ;BRANCH IF NO ERROR
105 057664  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
106 057670  112776  000147  000000          MOVB    #147,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
107 057676  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
108 057702  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
109 057704  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
110 057710  000240                          NOP
111 057712                         18$:
112
113 057712                         19$:
114
```

```
115
116 057712  062716  000004                ;AUGMENT RETURN ADDRESS IF ANY ERROR WAS FOUND
                            ADD     #4,(SP)          ;MOVE SP TO ERROR CALL
117 057716  105776  000000                TSTB    @(SP)            ;WAS AN ERROR DETECTED??
118 057722  001403                        BEQ     21$              ;NO!!
119 057724  062716  000004                ADD     #4,(SP)          ;YES - MOVE SP TO ERROR RETURN
120 057730  000402                        BR      23$
121 057732  162716  000004    21$:        SUB     #4,(SP) ;MOVE SP BACK TO NO ERROR RETURN
122 057736  000240    23$:        NOP
123 057740  000207                        RTS     PC               ;RETURN TO USER
124
```

```
                                        .SBTTL  DATA TRANSFER COMMAND STATUS CHECK SUBROUTINE

                                        ;THIS SUBROUTINE VERIFIES THE RESULTS OF ALL DATA TRANSFER COMMANDS
                                        ;USING STATUS STORED IN THE GET BUFFER AND TEST PARAMETERS
                                        ;STORED IN THE PUT BUFFER.  ERRORS ARE REPORTED BY WRITING
                                        ;THE ERROR NUMBER IN THE USERS ERROR CALL.

                                        ;USER'S SUBROUTINE CALL:
                                        ;(1)    JSR     PC,DTASTS
                                        ;(2)    BR      ??              RETURN HERE IF NO DATA ERRORS
                                        ;(3)    NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;(4)    ERROR                   SUB WRITES ERROR NUMBER
                                        ;(5)    JSR     PC,@(SP)+       USER RETURNS FOR MORE CHECKS
                                        ;(6)    ??                      SUB RETURNS HERE AFTER ALL
                                        ;                               ERRORS ARE REPORTED

057742                                  DTASTS:

                                        ;CLEAR USER'S ERROR CALL AND ERROR FLAGS
057742  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
057746  105076  000000                          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF TRAP
057752  162716  000004                          SUB     #4,(SP)         ;RESTORE SP TO NO ERROR
057756  005037  063336                          CLR     500$            ;CLEAR ERROR FLAGS

                                        ;REPORT ANY CONTROL BUS PARITY ERROR WHILE READING REMOTE REGISTERS,
                                        ;I.E., MCPE = 1
057762  032737  020000  001334                  BIT     #MCPE,RMCS1I    ;WAS THERE A PARITY ERROR??
057770  001422                                BEQ     10$             ;NO!!
057772  013737  001334  001140                MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
060000  042737  020000  001140                BIC     #MCPE,$GDDAT
060006  013737  001334  001142                MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
060014  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060020  112776  000013  000000                MOVB    #13,@(SP)       ;WRITE ERROR NUMBER
060026  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060032  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060034  000466                                BR      30$
060036                                  10$:

                                        ;REPORT ANY CONTROL BUS PARITY ERROR WHILE WRITTING REMOTE REGISTERS,
                                        ;I.E., PAR = 1 AND DPE = 0
060036  032737  000010  001350                  BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
060044  001435                                BEQ     20$             ;NO!!
060046  032737  000010  001376                BIT     #DPE,RMER2I     ;DATA PARITY ERROR ?
060054  001031                                BNE     20$             ;YES!!
060056  013737  001350  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
060064  042737  000010  001140                BIC     #PAR,$GDDAT
060072  013737  001350  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
060100  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
060104  112776  000050  000000                MOVB    #50,@(SP)       ;WRITE ERROR NUMBER
060112  032737  001000  001344                BIT     #MXF,RMCS2I     ;DID MXF GET SET??
060120  001003                                BNE     15$             ;YES!!
060122  112776  000274  000000                MOVB    #274,@(SP)      ;NO - CHANGE ERROR NUMBER
060130  162716  000002                  15$:  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060134  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060136  000425                                BR      30$

                                        ;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
060140                                  20$:
060140  032737  001000  001344                BIT     #MXF,RMCS2I     ;WAS 'MISSED TRANSFER' SET??
060146  001425                            BEQ     40$             ;NO!!
060150  013737  001344  001140            MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
060156  042737  001000  001140            BIC     #MXF,$GDDAT
060164  013737  001344  001142            MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
060172  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060176  112776  000275  000000            MOVB    #275,@(SP)      ;WRITE ERROR NUMBER
060204  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060210  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060212                                  30$:
                                        ;
                                        ;RESTORE SP TO NO ERROR RETURN AND BYPASS FURHTER STATUS CHECKING
060212  162716  000010                    SUB     #10,(SP)        ;MOVE SP TO NO ERROR
060216  000137  063310                    JMP     380$            ;SKIP TO END OF SUB
                                        ;
060222                                  40$:
                                        ;
                                        ;REPORT AN ERROR IF 'OPI' ERROR OCCURRED DUE TO 'MOL' = 0, OR IF 'OPI'
                                        ;AND 'MOL' ARE SET, BUT 'VV' IS RESET, INDICATING AN INTERMITTENT
                                        ;'MOL'
060222  032737  020000  001350            BIT     #OPI,RMER1I     ;IS 'OPI' SET??
060230  001447                            BEQ     60$             ;NO!!
060232  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
060240  042737  020000  001140            BIC     #OPI,$GDDAT
060246  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
060254  032737  010000  001346            BIT     #MOL,RMDSI      ;WAS MEDIUM OFF LINE??
060262  001404                            BEQ     45$             ;YES!!
060264  032737  000100  001346            BIT     #VV,RMDSI       ;WAS 'MOL' INTERMITTENT??
060272  001013                            BNE     50$             ;NO!!
060274  062716  000004            45$:    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060300  112776  000276  000000            MOVB    #276,@(SP)      ;WRITE ERROR NUMBER IN CALL
060306  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060312  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060314  162716  000010                    SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
060320  000413                            BR      60$
060322                                  50$:
                                        ;
                                        ;REPORT 'OPI' ERROR, WHICH IS DUE TO 'ON CYLINDER' NOT DROPPING OR
                                        ;'RUN' TIMEOUT (20 MS) OR SEARCH TIMEOUT (50 MS)
060322  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060326  112776  000277  000000            MOVB    #277,@(SP)      ;WRITE ERROR NUMBER IN CALL
060334  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060340  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060342  162716  000010                    SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
060346  000240                            NOP
060350                                  60$:
                                        ;
                                        ;LOOK FOR 'IVC' ERROR DURING COMMAND INITIATION
060350  032737  010000  001376            BIT     #IVC,RMER2I     ;WAS THERE AN 'IVC' ERROR??
060356  001432                            BEQ     70$             ;NO!!
                                        ;REPORT 'IVC' ERROR DUE TO 'VV' = 0, OR REPORT ERRONEOUS 'IVC' ERROR
060360  013737  001376  001140            MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
```

```
    115 060366  042737  010000  001140          BIC     #IVC,$GDDAT
    116 060374  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
    117 060402  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
    118 060406  112776  000300  000000          MOVB    #300,@(SP)      ;WRITE ERROR NUMBER IN CALL
    119 060414  032737  000100  001346          BIT     #VV,RMDSI       ;WAS VOLUME VALID??
    120 060422  001403                          BEQ     65$             ;NO!!
    121 060424  112776  000301  000000          MOVB    #301,@(SP)      ;CHANGE ERROR NUMBER
    122 060432  162716  000002          65$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    123 060436  004736                          JSR     PC,@(SP)+       ;REPORT ''IVC'' ERROR AND RETURN
    124 060440  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    125 060444                          70$:
    126
    127                                 ;SEE IF ''ILF'' OR 'RMR'' IS SET
    128 060444  032737  000007  001350          BIT     #ILR!ILF!RMR, RMER1I
    129 060452  001510                          BEQ     100$            ;NO ERRORS DETECTED
    130                                 ;REPORT AN ERROR IF ''ILR'' IS SET
    131 060454  032737  000002  001350          BIT     #ILR,RMER1I     ;WAS ''ILR'' DETECTED??
    132 060462  001424                          BEQ     80$             ;NO!!
    133 060464  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
    134 060472  042737  000002  001140          BIC     #ILR,$GDDAT
    135 060500  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    136 060506  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    137 060512  112776  000302  000000          MOVB    #302,@(SP)      ;WRITE ERROR NUMBER IN CALL
    138 060520  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    139 060524  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    140 060526  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    141 060532  000240                          NOP
    142 060534                          80$:
    143
    144                                 ;REPORT AN ERROR IF ''ILF'' IS SET
    145 060534  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS ''ILF'' DETECTED??
    146 060542  001424                          BEQ     90$             ;NO!!
    147 060544  013737  001350  001140          MOV     RMER1I,$GDDAT   ;_XPECTED STATUS
    148 060552  042737  000001  001140          BIC     #ILF,$GDDAT
    149 060560  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    150 060566  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    151 060572  112776  000303  000000          MOVB    #303,@(SP)      ;WRITE ERROR NUMBER IN CALL
    152 060600  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    153 060604  004736                          JSR     PC,@(SP)+       ;REPORT ERORR AND RETURN
    154 060606  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    155 060612  000240                          NOP
    156 060614                          90$:
    157
    158                                 ;REPORT AN ERROR IF 'RMR'' IS SET
    159 060614  032737  000004  001350          BIT     #RMR,RMER1I     ; AS 'RMR'' DETECTED??
    160 060622  001424                          BEQ     100$            ;NO!!
    161 060624  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
    162 060632  042737  000004  001140          BIC     #RMR,$GDDAT
    163 060640  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    164 060646  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    165 060652  112776  000304  000000          MOVB    #304,@(SP)      ;WRITE ERROR NUMBER IN CALL
    166 060660  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    167 060664  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    168 060666  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    169 060672  000240                          NOP
    170 060674                          100$:
    171                                 ;DETERMINE WHETHER OR NOT ''IAE'' SHOULD BE SET AND CHECK FOR ERROR
```

```
 172 060674  012737  002000  001140          MOV     #IAE,$GDDAT     ;SETUP FOR ''IAE'' = 1
 173 060702  052737  040000  063336          BIS     #SKI,500$       ;SETUP FOR ''SKI'' = 1
 174 060710  023727  001444  001'6'          CMP     RMDC0,#822.     ;GREATER THAN LAST CYLINDER ?
 175 060716  101025                          BHI     110$            ;YES - CYLINDER IS INVALID
 176 060720  042737  040000  063336          BIC     #SKI,500$       ;RESET SKI FLAG
 177
 178 060726  123737  001417  001333          CMPB    RMDA0+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
 179 060734  101016                          BHI     110$            ;YES - TRACK IS INVALID
 180
 181 060736  123727  001416  000035          CMPB    RMDA0,#29.      ;IS SECTOR > 29. ?
 182 060744  101410                          BLOS    105$            ;NO
 183 060746  032737  010000  001442          BIT     #FMT16,RMOF0    ;18 BIT FORMAT ?
 184 060754  001406                          BEQ     110$            ;YES - SECTOR IS INVALID FOR 18 BIT MODE
 185 060756  123727  001416  000037          CMPB    RMDA0,#31.      ;IS SECTOR > 31. ?
 186 060764  101002                          BHI     110$            ;YES - SECTOR IS INVALID
 187 060766  005037  001140          105$:   CLR     $GDDAT          ;''IAE'' SHOULD = 0
 188
 189 060772  013737  001350  001142  110$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
 190 061000  042737  175777  001142          BIC     #^CIAE,$BDDAT
 191 061006  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ''IAE'' STATUS OK??
 192 061014  001004                          BNE     115$            ;NO!!
 193 061016  042737  040000  063336          BIC     #SKI,500$       ;IAE OK - SKI SHOULD BE 0
 194 061024  000412                          BR      120$
 195 061026  062716  000004          115$:   ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 196 061032  112776  000305  000000          MOVB    #305,@(SP)      ;WRITE ERROR NUMBER
 197 061040  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 198 061044  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 199 061046  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 200 061052                          120$:
 201
 202                                 ;REPORT AN ERROR IF ''SKI'' IS SET AND ''IAE'' STATUS WAS OK
 203 061052  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
 204 061060  042737  137777  001142          BIC     #^CSKI,$BDDAT
 205 061066  013737  063336  001140          MOV     500$,$GDDAT     ;EXPECTED STATUS
 206 061074  042737  137777  001140          BIC     #^CSKI,$GDDAT
 207 061102  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS ''SKI'' SET??
 208 061110  001417                          BEQ     140$            ;NO!!
 209 061112  032737  040000  063336          BIT     #SKI,500$       ;WAS SKI CAUSED BY IAE = 0??
 210 061120  001032                          BNE     150$            ;YES - DON'T REPORT SKI
 211 061122  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 212 061126  112776  000306  000000          MOVB    #306,@(SP)      ;WRITE ERROR NUMBER
 213 061134  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 214 061140  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 215 061142  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 216 061146  000417                          BR      150$
 217
 218 061150                          140$:
 219
 220                                 ;REPORT AN ERROR IF SKI = 0 AND IAE WAS NOT DETECTED
 221 061150  032737  040000  063336          BIT     #SKI,500$       ;SHOULD SKI BE SET??
 222 061156  001413                          BEQ     150$            ;NO!!
 223 061160  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 224 061164  112776  000307  000000          MOVB    #307,@(SP)      ;WRITE ERROR NUMBER IN CALL
 225 061172  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 226 061176  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 227 061200  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
 228 061204  000240                          NOP
```

```
061206                                  150$:

                                        ;LOOK FOR 'LSC' OR 'LBC' OR 'DVC' IN ERROR REGISTER #2
061206  032737  006200  001376                BIT     #LSC!LBC!DVC,RMER2I
061214  001512                                BEQ     180$            ;NO ERRORS SET

                                        ;REPORT ANY DEVICE FAULT, I.E., 'DVC' = 1
061216  032737  000200  001376                BIT     #DVC,RMER2I     ;IS 'DVC' = 1??
061224  001424                                BEQ     160$            ;NO!!
061226  013737  001376  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
061234  042737  000200  001140                BIC     #DVC,$GDDAT
061242  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
061250  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
061254  112776  000310  000000                MOVB    #310,@(SP)      ;WRITE ERROR NUMBER IN CALL
061262  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061266  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061270  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
061274  000240                                NOP
061276                                  160$:

                                        ;REPORT LOSS OF BIT CLOCK, I.E.; 'LBC' = 1, IF 'MOL' = 1
061276  032737  002000  001376                BIT     #LBC,RMER2I     ;IS LBC SET??
061304  001430                                BEQ     170$            ;NO!!
061306  032737  010000  001346                BIT     #MOL,RMDSI      ;WAS LBC ERROR BY MOL = 0
061314  001424                                BEQ     170$            ;YES!!
061316  013737  001376  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
061324  042737  002000  001140                BIC     #LBC,$GDDAT
061332  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
061340  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
061344  112776  000311  000000                MOVB    #311,@(SP)      ;WRITE ERROR NUMBER IN CALL
061352  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061356  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061360  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
061364  000240                                NOP
061366                                  170$:

                                        ;REPORT LOS OF SYSTEM CLOCK, I.E., 'LSC' = 1
061366  032737  004000  001376                BIT     #LSC,RMER2I     ;IS 'LSC' = 1??
061374  001422                                BEQ     180$            ;NO!!
061376  013737  001376  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
061404  042737  004000  001140                BIC     #LSC,$GDDAT
061412  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
061420  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
061424  112776  000312  000000                MOVB    #312,@(SP)      ;WRITE ERROR NUMBER
061432  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061434  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
061440  000240                                NOP
061442                                  180$:

                                        ;LOOK FOR 'UNS' OR 'DTE' OR 'WLE' IN ERROR REGISTER #1
061442  032737  054000  001350                BIT     #UNS!DTE!WLE,RMER1I
061450  001527                                BEQ     220$            ;NO BITS SET
                                        ;REPORT 'UNS' ERROR IF 'DVC' = 0
061452  032737  040000  001350                BIT     #UNS,RMER1I     ;IS 'UNS' SET??
061460  001427                                BEQ     190$            ;NO!!
061462  032737  000200  001376                BIT     #DVC,RMER2I     ;WAS 'UNS' CAUSED BY 'DVC'??
061470  001023                                BNE     190$            ;YES!!
```

```
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #UNS,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #313,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                190$:

                                ;REPORT ANY DRIVE TIMING ERROR, I.E., 'DTE' = 1
                                        BIT     #DTE,RMER1I     ;IS DTE SET??
                                        BEQ     200$            ;NO!!
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #DTE,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #314,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                200$:

                                ;REPORT AN ERROR IF WRITE LOCK ERROR IS SET.  SEE IF DRIVE IS NOT
                                ;WRITE PROTECTED, OR IF FUNCTION WAS NOT A WRITE
                                        BIT     #WLE,RMER1I     ;WAS 'WLE' SET??
                                        BEQ     220$            ;NO!!
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIS     #WLE,$GDDAT
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #315,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        BIT     #WRL,RMDSI      ;WAS DRIVE WRITE PROTECTED??
                                        BEQ     205$            ;NO!!
                                        BIT     #BIT3,RMCS1O    ;WAS COMMAND A WRITE??
                                        BEQ     210$            ;YES!!
                                205$:   MOVB    #316,@(SP)      ;CHANGE ERROR NUMBER
                                        BIC     #WLE,$GDDAT
                                210$:   SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR

                                220$:

                                ;OMIT DATA ERROR CHECKS IF ANY PREVIOUS ERRORS HAVE BEEN DETECTED
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                        TSTB    @(SP)           ;WAS ERROR DETECTED??
                                        BEQ     225$            ;NO - DO DATA CHECKS
                                        SUB     #4,(SP)         ;RESTORE SP
                                        JMP     340$            ;SKIP DATA CHECKS
                                225$:   SUB     #4,(SP)         ;RESTORE SP

                                ;CHECK HEADER ERRORS IF FUNCTION WAS NOT WRITE HEADER AND DATA, AND
                                ;IF HEADER COMPARE IS NOT INHIBITED
                                        MOV     RMCS1O,510$     ;STRIP AND STORE FUNCTION CODE
                                        BIC     #^CFNCMSK,510$
                                        CMP     #WH!GO,510$     ;WAS FUNCTION WRITE HEADER & DATA??
```

```
                                        BEQ     250$            ;YES - SKIP HEADER CHECKS
062002  032737  002000  001366        BIT     #HCI,RMOFI      ;WAS HCI SET??
062010  001106                        BNE     250$            ;YES - SKIP HEADER CHECKS

                                ;SEE IF ANY HEADER ERRORS ARE SET, I.E., 'FER' OR 'HCRC' OR 'HCE'
062012  032737  000620  001350        BIT     #HCRC!FER!HCE,RMER1I
062020  001533                        BEQ     270$            ;NO ERRORS SET

                                ;REPORT HEADER CRC ERROR IF SET
062022  032737  000400  001350        BIT     #HCRC,RMER1I    ;WAS HCRC SET??
062030  001422                        BEQ     230$            ;NO!!
062032  013737  001350  001140        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
062040  042737  000400  001140        BIC     #HCRC,$GDDAT
062046  013737  001350  001142        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
062054  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
062060  112776  000317  000000        MOVB    #317,@(SP)      ;WRITE ERROR NUMBER
062066  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
062072  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
062074  000501                        BR      260$
062076                          230$:

                                ;REPORT FORMAT ERROR IF SET
062076  032737  000020  001350        BIT     #FER,RMER1I     ;WAS 'FER' SET??
062104  001422                        BEQ     240$            ;NO!!
062106  013737  001350  001140        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
062114  042737  000020  001140        BIC     #FER,$GDDAT
062122  013737  001350  001142        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
062130  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
062134  112776  000320  000000        MOVB    #320,@(SP)      ;WRITE ERROR NUMBER
062142  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
062146  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
062150  000453                        BR      260$
062152                          240$:

                                ;REPORT HEADER COMPARE ERROR IF SET
062152  032737  000200  001350        BIT     #HCE,RMER1I     ;WAS 'HCE' SET??
062160  001453                        BEQ     270$            ;NO!!
062162  013737  001350  001140        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
062170  042737  000200  001140        BIC     #HCE,$GDDAT
062176  013737  001350  001142        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
062204  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
062210  112776  000321  000000        MOVB    #321,@(SP)      ;WRITE ERROR NUMBER
062216  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
062222  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
062224  000425                        BR      260$
                                ;THERE SHOULD BE NO HEADER ERRORS BECAUSE
                                ;       .COMMAND WAS WRITE HEADER AND DATA, OR
                                ;       .HEADER COMPARE INHIBIT WAS SET
062226  032737  000620  001350  250$:   BIT     #HCE!FER!HCRC,RMER1I
062234  001425                        BEQ     270$            ;NO ERRORS WERE SET
062236  013737  001350  001140        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
062244  042737  000620  001140        BIC     #HCE!FER!HCRC,$GDDAT
062252  013737  001350  001142        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
062260  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
062264  112776  000322  000000        MOVB    #322,@(SP)      ;WRITE ERROR NUMBER
062272  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
062276  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
```

```
                                260$:   SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        JMP     340$            ;OMIT FURTHER DATA CHECKS

                                270$:

                                ;IF COMMAND WAS A WRITE COMMAND, GO DO WRITE ERROR CHECKS, OTHERWISE
                                ;DO READ ERROR CHECKS
062310  032737  000010  063340            BIT     #BIT3,510$      ;WAS THIS A WRITE COMMAND?
062316  001002                            BNE     275$            ;NO!!
062320  000137  062536                    JMP     310$            ;GO DO WRITE STATUS CHECK
062324                          275$:

                                ;REPORT DATA CHECK IF SET
062324  032737  100000  001350            BIT     #DCK,RMER1I     ;DATA CHECK ERROR??
062332  001450                            BEQ     290$            ;NO!!
062334  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
062342  042737  100000  001140            BIC     #DCK,$GDDAT
062350  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
062356  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
062362  112776  000323  000000            MOVB    #323,@(SP)      ;WRITE ERROR NUMBER
062370  032737  004000  001366            BIT     #ECI,RMOFI      ;WAS ECC CORRECTION DISABLED??
062376  001021                            BNE     280$            ;YES!!
062400  112776  000324  000000            MOVB    #324,@(SP)      ;CHANGE TO RECOVERABLE ERROR
062406  032737  000100  001350            BIT     #ECH,RMER1I     ;IS ERROR RECOVERABLE??
062414  001007                            BNE     276$            ;NO !!
                                ;DO NOT REPORT RECOVERABLE ERROR IF READ COMMAND
062416  032737  000020  063340            BIT     #BIT4,510$      ;WAS THIS A READ COMMAND ??
062424  001406                            BEQ     280$            ;NO !!
062426  162716  000004                    SUB     #4,(SP)         ;RESTORE SP
062432  000410                            BR      290$            ;SKIP ERROR - DATA WILL BE CORRECTED
062434  112776  000325  000000  276$:     MOVB    #325,@(SP)      ;CHANGE TO NON RECOVERABLE
062442  162716  000002          280$:     SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
062446  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
062450  162716  000010                    SUB     #10,(SP)        ;RESTORE SP TO NO ERROR

062454                          290$:

                                ;REPORT DATA BUS PARITY ERROR IF SET, I.E., MDPE = 1
062454  032737  000400  001344            BIT     #MDPE,RMCS2I    ;PARITY ERROR SET??
062462  001423                            BEQ     300$            ;NO!!
062464  013737  001344  001140            MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
062472  042737  000400  001140            BIC     #MDPE,$GDDAT
062500  013737  001344  001142            MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
062506  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
062512  112776  000326  000000            MOVB    #326,@(SP)      ;WRITE ERROR NUMBER
062520  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
062524  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
062526  162716  000010                    SUB     #10,(SP)        ;MOVE SP TO NO ERROR
062532  000137  062750          300$:     JMP     340$            ;SKIP WRITE STATUS CHECK

062536                          310$:

                                ;TEST TO SEE THAT OFFSET MODE WAS RESET; REPORT ERROR IF 'OM' = 1
062536  032737  000001  001346            BIT     #OM,RMDSI       ;IS OFFSET ON??
062544  001423                            BEQ     320$            ;NO
062546  013737  001346  001140            MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
062554  042737  000001  001140            BIC     #OM,$GDDAT
```

```
                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                        MOVB    #327,@(SP)      ;WRITE ERROR NUMBER IN CALL
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                320$:

                ;TEST FOR DATA BUS PARITY ERROR; REPORT ERROR IF 'DPE'' = 1
                        BIT     #DPE,RMER2I     ;DATA PARITY ERROR??
                        BEQ     330$            ;NO!!
                        MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
                        BIC     #DPE,$GDDAT
                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                        MOVB    #330,@(SP)      ;WRITE ERROR NUMBER
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                330$:

                ;TEST FOR WRITE CLOCK FAILURE; REPORT ERROR IF 'WCF'' = 1
                        BIT     #WCF,RMER1I     ;IS 'WCF'' SET??
                        BEQ     340$            ;NO!!
                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                        BIC     #WCF,$GDDAT
                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                        MOVB    #331,@(SP)      ;WRITE ERROR NUMBER
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                340$:

                ;REPORT 'DATA LATE'' ERROR IF 'DLT'' = 1
                        BIT     #DLT,RMCS2I     ;IS 'DLT'' SET??
                        BEQ     350$            ;NO!!
                        MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
                        BIC     #DLT,$GDDAT
                        MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                        MOVB    #332,@(SP)      ;WRITE ERROR NUMBER
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                350$:
                ;LOOK FOR UNEXPECTED CHANGES IN DRIVE STATUS
                        MOV     RMDSI,-(SP)     ;STACK DRIVE STATUS
                        BIC     #^C<PIP!MOL!VV>,(SP) ;CLEAR DONT CARES
                        CMP     #MOL!VV,(SP)+   ;IS DRIVE STATUS OK??
                        BEQ     380$            ;YES!!

                ;REPORT ERROR IF POSITIONING IN PROGRESS AND NO SEEK INCOMPLETE ERROR,
                ;I.E, PIP = 1 AND SKI = 0
                        BIT     #PIP,RMDSI      ;IS 'PIP'' SET??
                        BEQ     360$            ;NO!!
                        BIT     #SKI,RMER2I     ;WAS ''SKI'' ERROR REPORTED??
```

```
514 063062  001024                        BNE     360$            ;YES-DONT REPORT PIP
515 063064  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
516 063072  042737  020000  001140        BIC     #PIP,$GDDAT
517 063100  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
518 063106  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
519 063112  112776  000333  000000        MOVB    #333,@(SP)      ;WRITE ERROR NUMBER
520 063120  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
521 063124  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
522 063126  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
523 063132  000240                        NOP
524 063134                          360$:
525
526                                 ;REPORT ERROR IF MEDIUM IS NOT ON LINE AND OPI ERROR WAS NOT
527                                 ;REPORTED, I.E., MOL = OPI = 0
528 063134  032737  010000  001346        BIT     #MOL,RMDSI      ;IS MEDIUM ON LINE??
529 063142  001027                        BNE     370$            ;YES!!
530 063144  032737  020000  001350        BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
531 063152  001023                        BNE     370$            ;YES!!
532 063154  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
533 063162  052737  010000  001140        BIS     #MOL,$GDDAT
534 063170  013737  001346  001142        MOV     RMDS ,$BDDAT    ;RECEIVED STATUS
535 063176  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
536 063202  112776  000334  000000        MOVB    #334,@(SP)      ;WRITE ERROR NUMBER
537 063210  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
538 063214  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
539 063216  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
540 063222                          370$:
541
542                                 ;REPORT ERROR IF VOLUME IS NOT VALID AND "IVC" ERROR WAS NOT
543                                 ;REPORTED, I.E., VV = IVC = 0
544 063222  032737  000100  001346        BIT     #VV,RMDSI       ;IS VOLUME VALID??
545 063230  001027                        BNE     380$            ;YES!!
546 063232  032737  010000  001376        BIT     #IVC,RMER2I     ;WAS IVC ERROR REPORTED??
547 063240  001033                        BNE     390$            ;YES!!
548 063242  013737  001346  001140        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
549 063250  052737  000100  001140        BIS     #VV,$GDDAT
550 063256  013737  001346  001142        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
551 063264  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
552 063270  112776  000335  000000        MOVB    #335,@(SP)      ;WRITE ERROR NUMBER
553 063276  162716  000002                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
554 063302  004736                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
555 063304  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
556 063310                          380$:
557
558                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS FOUND
559 063310  062716  000004                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
560 063314  105776  000000                TSTB    @(SP)           ;ANY ERROR??
561 063320  001403                        BEQ     390$            ;NO!!
562 063322  062716  000004                ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
563 063326  000402                        BR      400$
564 063330  162716  000004          390$: SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
565
566 063334  000207                  400$: RTS     PC              ;RETURN TO USER
567
568 063336  000000                  500$: .WORD                   ;ERROR FLAGS
569 063340  000000                  510$: .WORD                   ;TEMPORARY STORAGE
570
```

```
                                .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

063342                          STCDRVSTS:

                                ;CLEAR USER'S ERROR CALL
063342  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
063346  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
063352  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR RETURN
                                ;SEE IF 'MOL' = 'VV' = 1, AND 'PIP' = 0
063356  013746  001346                  MOV     RMDSI,-(SP)     ;PUT DRIVE STATUS ON STACK
063362  042716  147677                  BIC     #^C<PIP!MOL!VV>,(SP)
063366  022726  010100                  CMP     #MOL!VV,(SP)+   ;ARE MOL,VV AND PIP O.K.??
063372  001524                          BEQ     30$             ;YES!!

                                ;REPORT AN ERROR IF MOL = 0 AND 'OPI' = 0
063374  032737  010000  001346          BIT     #MOL,RMDSI      ;IS MOL ON ??
063402  001030                          BNE     10$             ;YES!!
063404  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS 'OPI' SET??
063412  001024                          BNE     10$             ;YES-DONT REPORT 'MOL' = 0
063414  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
063422  052737  010000  001140          BIS     #MOL,$GDDAT
063430  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
063436  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
063442  112776  000207  000000          MOVB    #207,@(SP)      ;WRITE ERROR NUMBER IN CALL
063450  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
063454  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
063456  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
063462  000240                          NOP
063464                          10$:

                                ;REPORT AN ERROR IF VOLUME VALID IS NOW ZERO AND 'IVC' = 0
063464  032737  000100  001346          BIT     #VV,RMDSI       ;IS 'VV' = 0??
063472  001030                          BNE     20$             ;NO!!
```

```
     58 063474  032737  010000  001376          BIT     #IVC,RMER2I     ;WAS ''IVC'' SET??
     59 063502  001024                          BNE     20$             ;YES-DONT REPORT ''VV'' = 0
     60 063504  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
     61 063512  052737  000100  001346          BIS     #VV,RMDSI
     62 063520  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
     63 063526  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     64 063532  112776  000210  000000          MOVB    #210,@(SP)      ;WRITE ERROR NUMBER IN CALL
     65 063540  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     66 063544  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     67 063546  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     68 063552  000240                          NOP
     69 063554                          20$:
     70
     71                                 ;REPORT AN ERROR IF DRIVE IS OFF CYLINDER AND ''SKI'' = 0
     72 063554  032737  020000  001346          BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
     73 063562  001430                          BEQ     30$             ;NO!!
     74 063564  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS ''SKI'' SET??
     75 063572  001024                          BNE     30$             ;YES-DONT REPORT ''PIP'' = 1
     76 063574  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
     77 063602  042737  020000  001140          BIC     #PIP,$GDDAT
     78 063610  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
     79 063616  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     80 063622  112776  000211  000000          MOVB    #211,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
     81 063630  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     82 063634  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     83 063636  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     84 063642  000240                          NOP
     85 063644                          30$:
     86
     87                                 ;SEE IF ''SKI'' = ''DVC'' = 0
     88 063644  013746  001376                  MOV     RMER2I,-(SP)    ;PUT ERROR REG 2 ON STACK
     89 063650  042726  137577                  BIC     #^C<SKI!DVC>,(SP)+
     90 063654  001460                          BEQ     60$             ;BRANCH IF NO ERROR
     91 063656                          40$:
     92
     93                                 ;REPORT AN ERROR IF THERE IS A DEVICE FAULT
     94 063656  032737  000200  001376          BIT     #DVC,RMER2I     ;ANY DEVICE FAULT??
     95 063664  001424                          BEQ     50$             ;NO!!
     96 063666  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
     97 063674  042737  000200  001140          BIC     #DVC,$GDDAT
     98 063702  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     99 063710  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S CALL
    100 063714  112776  000212  000000          MOVB    #212,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
    101 063722  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    102 063726  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    103 063730  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    104 063734  000240                          NOP
    105 063736                          50$:
    106
    107                                 ;REPORT AN ERROR IF ''SKI'' = 1
    108 063736  032737  040000  001376          BIT     #SKI,RMER2I     ;IS THERE A SEEK INCOMPLETE ERROR
    109 063744  001424                          BEQ     60$             ;NO!!
    110 063746  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
    111 063754  042737  040000  001140          BIC     #SKI,$GDDAT
    112 063762  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
    113 063770  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    114 063774  112776  000213  000000          MOVB    #213,@(SP)      ;WRITE ERROR NUMBER IN USER'S ERROR CALL
```

```
    115 064002  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    116 064006  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    117 064010  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    118 064014  000240                          NOP
    119 064016                          60$:
    120
    121                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
    122 064016  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    123 064022  105776  000000                  TSTB    @(SP)           ;WAS AN ERROR DETECTED??
    124 064026  001403                          BEQ     70$             ;NO!!
    125 064030  062716  000004                  ADD     #4,(SP)         ;YES - MOVE SP TO USER'S ERROR RETURN
    126 064034  000402                          BR      80$
    127 064036  162716  000004          70$:    SUB     #4,(SP)         ;NO - MOVE SP TO NO ERROR RETURN
    128 064042  000240                  80$:    NOP
    129 064044  000207                          RTS     PC              ;RETURN TO USER
    130
```

```
                                        .SBTTL  STOP AND SHUTDOWN SUBROUTINES
                                
 064046                                 STOP:
                                
                                        ;DROP PRIORITY TO ALLOW CONSOLE INTERRUPT
 064046  012746  000140                         MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
 064052  012746  064060                         MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
 064056  000002                                 RTI                     ;;POP NEW PC AND PS
 064060                                 64$:
 064060  000240                                 NOP
                                
                                        ;RAISE PRIORITY TO INHIBIT CONSOLE INTERRUPT
 064062  012746  000300                         MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
 064066  012746  064074                         MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
 064072  000002                                 RTI                     ;;POP NEW PC AND PS
 064074                                 65$:
 064074  000207                                 RTS     PC              ;CONTINUE
                                
 064076  005737  001326                 SHUT:   TST     CTLFG           ;HALT ?
 064102  001002                                 BNE     5$              ;BR IF YES
 064104  000137  007764                         JMP     READY           ;CONTINUE
 064110  005737  000042                 5$:     TST     @#42            ;ANY MONITOR PRESENT ?
 064114  001015                                 BNE     10$             ;BR IF YES
 064116  104401  064124                         TYPE    ,65$            ;;TYPE ASCIZ STRING
 064122  000410                                 BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF><07>/TEST HALTED/<CRLF>
 064144                                 64$:
 064144  000137  005420                         JMP     START           ;GO TO START
 064150  000137  037460                 10$:    JMP     $EOP            ;RETURN CONTROL TO MONITOR
                                
 064154  012737  177777  001326       SHUT2:  MOV     #-1,CTLFG       ;SET THE CONTROL-C FLAG
 064162  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

 064164                                 $SAVREG:
 064164  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 064166  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 064170  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 064172  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 064174  010446                                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 064176  010546                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 064200  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 064204  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 064210  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 064214  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 064220  000002                               RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
 064222                                 $RESREG:
 064222  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 064226  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 064232  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 064236  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 064242  012605                               MOV     (SP)+,R5        ;;POP STACK INTO R5
 064244  012604                               MOV     (SP)+,R4        ;;POP STACK INTO R4
 064246  012603                               MOV     (SP)+,R3        ;;POP STACK INTO R3
 064250  012602                               MOV     (SP)+,R2        ;;POP STACK INTO R2
 064252  012601                               MOV     (SP)+,R1        ;;POP STACK INTO R1
 064254  012600                               MOV     (SP)+,R0        ;;POP STACK INTO R0
 064256  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

 064260  010146                         $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 064262  016601  000006                         MOV     6(SP),R1        ;;GET THE INPUT NUMBER
 064266  000261                               SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
```

```
                                                                    N 4
064270  112737  000060  064332  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII '0'.
064276  006101                          ROL     R1              ;;GET THIS BIT
064300  001406                          BEQ     2$              ;;DONE?
064302  105537  064332                  ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
064306  104401  064332                  TYPE    ,$BIN           ;;GO TYPE THIS BIT
064312  000241                          CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
064314  000765                          BR      1$              ;;GO DO THE NEXT BIT
064316  012601                  2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
064320  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
064326  012616                          MOV     (SP)+,(SP)
064330  000002                          RTI                     ;;RETURN TO USER
064332     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

064334                          $TYPDS:
064334  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
064336  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
064340  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
064342  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
064344  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
064346  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
064352  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
064356  100004                          BPL     1$              ;;BR IF INPUT IS POS.
064360  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
064362  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
064370  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
064372  012703  064550                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
064376  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
064402  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
064404  016001  064540                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
064410  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
064412  002402                          BLT     4$              ;;BR IF DONE
064414  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
064416  000774                          BR      3$
064420  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
064422  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
064424  001002                          BNE     5$              ;;FALL THROUGH IF 0
064426  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
064430  100407                          BMI     7$              ;;BR IF YES
064432  106316                  5$:     ASLB    (SP)            ;;MSD?
064434  103003                          BCC     6$              ;;BR IF NO
064436  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
064444  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
064450  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
064454  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
064456  005720                          TST     (R0)+           ;;JUST INCREMENTING
064460  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
```

```
064464  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
064466  003002                          BGT     8$              ;;GO TO EXIT
064470  010502                          MOV     R5,R2           ;;GET THE LSD
064472  000764                          BR      6$              ;;GO CHANGE TO ASCII
064474  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
064476  100003                          BPL     9$              ;;BR IF NO
064500  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
064506  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
064510  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
064512  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
064514  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
064516  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
064520  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
064522  104401  064550                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
064526  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
064534  012616                          MOV     (SP)+,(SP)
064536  000002                          RTI                     ;;RETURN TO USER
064540  023420                  $DTBL:  10000.
064542  001750                          1000.
064544  000144                          100.
064546  000012                          10.
064550                          $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

064560  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
064564  116637  000001  065003          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
064572  112637  065005                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
064576  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
064602  000406                          BR      $TYPON
064604  112737  000001  065003  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
064612  112737  000006  065005          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
064620  112737  000005  065002  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
064626  010346                          MOV     R3,-(SP)        ;;SAVE R3
064630  010446                          MOV     R4,-(SP)        ;;SAVE R4
```

```
064632  010546                          MOV     R5,-(SP)        ;;SAVE R5
064634  113704  065005                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
064640  005404                          NEG     R4
064642  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
064646  110437  065004                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
064652  113704  065003                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
064656  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
064662  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
064664  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO ''C''
064666  000404                          BR      3$              ;;GO DO MSB
064670  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
064672  006105                          ROL     R5
064674  006105                          ROL     R5
064676  010503                          MOV     R5,R3
064700  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
064702  105337  065004                  DECB    $OMODE          ;;TYPE THIS DIGIT?
064706  100016                          BPL     7$              ;;BR IF NO
064710  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
064714  001002                          BNE     4$              ;;TEST FOR 0
064716  005704                          TST     R4              ;;SUPPRESS THIS 0?
064720  001403                          BEQ     5$              ;;BR IF YES
064722  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
064724  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
064730  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
064734  110337  065000                  MOVB    R3,8$           ;;SAVE FOR TYPING
064740  104401  065000                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
064744  105337  065002          7$:     DECB    $OCNT           ;;COUNT BY 1
064750  003347                          BGT     2$              ;;BR IF MORE TO DO
064752  002402                          BLT     6$              ;;BR IF DONE
064754  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
064756  000744                          BR      2$              ;;GO DO THE LAST DIGIT
064760  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
064762  012604                          MOV     (SP)+,R4        ;;RESTORE R4
064764  012603                          MOV     (SP)+,R3        ;;RESTORE R3
064766  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
064774  012616                          MOV     (SP)+,(SP)
064776  000002                          RTI                     ;;RETURN
065000     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
065001     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
065002     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
065003     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
065004  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
```

```
                                ;*
065006  105737  001173        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
065012  100002                        BPL     1$              ;;BR IF YES
065014  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
065016  000430                        BR      3$              ;;LEAVE
065020  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
065022  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
065026  122737  000001  001242        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
065034  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
065036  132737  000100  001243        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
065044  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
065046  010037  065056                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
065052  004737  067770                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
065056  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
065060  132737  000040  001243  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
065066  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
065070  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
065072  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
065074  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
065076  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
065100  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
065104  000002                        RTI                     ;;RETURN
065106  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
065112  001430                        BEQ     8$
065114  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
065120  001006                        BNE     5$
065122  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
065124  104401                        TYPE                    ;;TYPE A CR AND LF
065126  001217                        $CRLF
065130  105037  065336                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
065134  000755                        BR      2$              ;;GET NEXT CHARACTER
065136  004737  065220        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
065142  123726  001172        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
065146  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
065150  013746  001170                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                              ;;AND THE NULL CHAR.
065154  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
065160  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
065162  004737  065220                JSR     PC,$TYPEC       ;;GO TYPE A NULL
065166  105337  065336                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
065172  000770                        BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

065174  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
065200  004737  065220        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
065204  132737  000007  065336        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
065212  001372                        BNE     9$              ;;TAB STOP
065214  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
065216  000724                        BR      2$              ;;GET NEXT CHARACTER
065220                          $TYPEC:
065220  105777  113734                TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
065224  100022                        BPL     10$             ;;BR IF NOT
065226  017746  113730                MOV     @$TKB,-(SP)     ;;GET CHAR
065232  042716  177600                BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
065236  122716  000023                CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
```

```
065242  001012                          BNE     102$            ;;BR IF NOT
065244                          101$:
065244  105777  113710                  TSTB    @$TKS           ;;WAIT FOR CHAR
065250  100375                          BPL     101$
065252  117716  113704                  MOVB    @$TKB,(SP)      ;;GET CHAR
065256  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
065262  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
065266  001366                          BNE     101$            ;;BR IF NOT
065270                          102$:
065270  005726                          TST     (SP)+           ;;FIX STACK
065272                          10$:
065272  105777  113666                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
065276  100375                          BPL     10$
065300  116677  000002  113660          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
065306  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
065314  001003                          BNE     1$              ;;BRANCH IF NO
065316  105037  065336                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
065322  000406                          BR      $TYPEX          ;;EXIT
065324  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
065332  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
065334  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
065336  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
065340  000207                  $TYPEX: RTS     PC

    6                           .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

065342                          $SCOPE:
065342  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
065344  004737  064046                  JSR     PC,STOP
065350  032777  040000  113576  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
065356  001131                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
065360  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A 'NOP' (NOP=240)
065362  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
065366  012737  065406  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
065374  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
065400  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
065404  000500                          BR      $SVLAD          ;;GO TO THE NEXT TEST
065406  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
065410  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
065414  000440                          BR      7$              ;;LOOP ON THE PRESENT TEST
065416                          6$:;#####END OF CODE FOR THE XOR TESTER#####
065416  032777  000400  113530          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
065424  001421                          BEQ     2$              ;;BR IF NO
```

```
065426  005046                          CLR     -(SP,                   ;;CLEAR A TEMP. LOCATION
065430  117716  113520                  MOVB    @SWR,(SP)               ;;PICKUP THE DESIRED TEST NUMBER
065434  001414                          BEQ     8$                      ;;BRANCH IF BAD TEST NUMBER IN SWR
065436  022716  000027                  CMP     #27,(SP)                ;;CHECK THE NUMBER IN THE SWR
065442  002411                          BLT     8$                      ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
065444  011637  001116                  MOV     (SP),$TSTNM             ;;UPDATE THE TEST NUMBER
065450  005316                          DEC     (SP)                    ;;BACKUP BY ONE
065452  006316                          ASL     (SP)                    ;;SCALE THE TEST NUMBER AS AN INDEX
065454  062716  065660                  ADD     #$SW08TBL,(SP)          ;;FORM THE ADDRESS OF TEST POINTER
065460  013637  001122                  MOV     @(SP)+,$LPADR           ;;SET LOOP ADDRESS TO DESIRED TEST
065464  000466                          BR      $OVER                   ;;GO LOOP ON THE TEST
065466  005726                  8$:     TST     (SP)+                   ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
065470  105737  001117          2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
065474  001421                          BEQ     3$                      ;;BR IF NO
065476  123737  001131  001117          CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
065504  101015                          BHI     3$                      ;;BR IF NO
065506  032777  001000  113440          BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
065514  001404                          BEQ     4$                      ;;BR IF NO
065516  013737  001124  001122  7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
065524  000446                          BR      $OVER
065526  105037  001117          4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
065532  005037  001206                  CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
065536  000415                          BR      1$                      ;;ESCAPE TO THE NEXT TEST
065540  032777  004000  113406  3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
065546  001011                          BNE     1$                      ;;BR IF YES
065550  005737  001230                  TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
065554  001406                          BEQ     1$                      ;;      INHIBIT ITERATIONS
065556  005237  001120                  INC     $ICNT                   ;;INCREMENT ITERATION COUNT
065562  023737  001206  001120          CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
065570  002024                          BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
065572  012737  000001  001120  1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
065600  013737  065656  001206          MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
065606  105237  001116          $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
065612  113737  001116  001226          MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
065620  011637  001122                  MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
065624  011637  001124                  MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
065630  005037  001210                  CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
065634  112737  000001  001131          MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
065642  013777  001116  113306  $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
065650  013716  001122                  MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
065654  000002                          RTI                             ;;FIXES PS
065656  000012          $MXCNT: 10.                                     ;;MAX. NUMBER OF ITERATIONS
065660                  $SW08TBL:
065660  010132                  .WORD   TST1+2                          ;;STARTING ADDRESS OF TEST 1
065662  010316                  .WORD   TST2+2                          ;;STARTING ADDRESS OF TEST 2
065664  010502                  .WORD   TST3+2                          ;;STARTING ADDRESS OF TEST 3
065666  010704                  .WORD   TST4+2                          ;;STARTING ADDRESS OF TEST 4
065670  011712                  .WORD   TST5+2                          ;;STARTING ADDRESS OF TEST 5
065672  012670                  .WORD   TST6+2                          ;;STARTING ADDRESS OF TEST 6
065674  014064                  .WORD   TST7+2                          ;;STARTING ADDRESS OF TEST 7
065676  015072                  .WORD   TST10+2                         ;;STARTING ADDRESS OF TEST 10
065700  016050                  .WORD   TST11+2                         ;;STARTING ADDRESS OF TEST 11
065702  017246                  .WORD   TST12+2                         ;;STARTING ADDRESS OF TEST 12
065704  020252                  .WORD   TST13+2                         ;;STARTING ADDRESS OF TEST 13
065706  021476                  .WORD   TST14+2                         ;;STARTING ADDRESS OF TEST 14
065710  022704                  .WORD   TST15+2                         ;;STARTING ADDRESS OF TEST 15
065712  023726                  .WORD   TST16+2                         ;;STARTING ADDRESS OF TEST 16
```

```
065714  025574                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
065716  027200                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
065720  030220                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
065722  031136                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
065724  032174                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
065726  033200                          .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
065730  034206                          .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
065732  035214                          .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
065734  036242                          .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
    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

065736                          $ERROR:
065736  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
065740  105237  001117          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
065744  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
065746  013777  001116  113202          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
065754  032777  002000  113172          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
065762  001402                          BEQ     1$              ;;NO - SKIP
065764  104401  001212                  TYPE    ,$BELL          ;;RING BELL
065770  005237  001126          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
065774  011637  001132                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
066000  162737  000002  001132          SUB     #2,$ERRPC
066006  117737  113120  001130          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
066014  032777  020000  113132          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
066022  001004                          BNE     20$             ;;SKIP TYPEOUTS
066024  004737  037670                  JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
066030  104401  001217                  TYPE    ,$CRLF
066034                          20$:
066034  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
066042  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
066044  113737  001130  066056          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
066052  004737  070000                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
066056     000                  21$:    .BYTE   0
066057     000                          .BYTE   0
066060  000777                  22$:    BR      22$             ;;APT ERROR LOOP
066062  005777  113066          2$:     TST     @SWR            ;;HALT ON ERROR
066066  100002                          BPL     3$              ;;SKIP IF CONTINUE
066070  000000                          HALT                    ;;HALT ON ERROR!
066072  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
066074  032777  001000  113052  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
066102  001402                          BEQ     4$              ;;BR IF NO
066104  013716  001124                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
066110  005737  001210          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
066114  001402                          BEQ     5$              ;;BR IF NONE
066116  013716  001210                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
066122                          5$:
066122  022737  037650  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
066130  001001                          BNE     6$              ;;BRANCH IF NO
066132  000000                          HALT                    ;;YES
066134                          6$:
066134  000002                          RTI                     ;;RETURN
      8                         .SBTTL  TTY INPUT ROUTINE

                                ;;**********************************************************************
                                .ENABL  LSB
066136  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
066140  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
066142  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
066144                          $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        066145                  $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
                                ;
066146  005037  066136          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
066152  012737  066144  066140          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
066160  013737  066140  066142          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
066166  012737  066216  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
066174  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;'BR' LEVEL 4
066202  005777  112754                  TST     @$TKB           ;;CLEAR DONE FLAG
066206  012777  000100  112744          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
066214  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)
                                ;
066216  117746  112740          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
066222  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
066226  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
066232  001007                          BNE     1$              ;;BRANCH IF NO
066234  104401  067332                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
066240  004737  066146                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
066244  005726                          TST     (SP)+           ;;CLEAN UP STACK
066246  000137  064154                  JMP     SHUT2           ;;CONTROL C RESTART
066252  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
066256  001004                          BNE     2$              ;;BRANCH IF NO
066260  022737  000176  001154          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
066266  001500                          BEQ     6$              ;;GO TO SWR CHANGE

066270                          2$:
066270  022737  000001  066136          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
066276  001004                          BNE     3$              ;;BRANCH IF NO
```

```
066300  104401  001212          TYPE    ,$BELL          ;;RING THE TTY BELL
066304  005726                  TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
066306  000451                  BR      5$              ;;EXIT
066310  021627  000023  3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
066314  001021                  BNE     32$             ;;BRANCH IF NO
066316  005077  112636          CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
066322  005726                  TST     (SP)+           ;;CLEAN CHAR OFF STACK
066324  105777  112630  31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
066330  100375                  BPL     31$             ;;LOOP UNTIL ITS THERE
066332  117746  112624          MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
066336  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
066342  022627  000021          CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
066346  001366                  BNE     31$             ;;BRANCH IF NO
066350  012777  000100  112602  MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
066356  000002                  RTI                     ;;RETURN
066360  005237  066136  32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
066364  021627  000140          CMP     (SP),#140       ;;IS IT UPPER CASE?
066370  002405                  BLT     4$              ;;BRANCH IF YES
066372  021627  000175          CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
066376  003002                  BGT     4$              ;;BRANCH IF YES
066400  042716  000040          BIC     #40,(SP)        ;;MAKE IT UPPER CASE
066404  112677  177530  4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
066410  005237  066140          INC     $TKQIN          ;;UPDATE THE POINTER
066414  023727  066140  066145  CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
066422  001003                  BNE     5$              ;;BRANCH IF NO
066424  012737  066144  066140  MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
066432  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.
066434  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
066442  001124                  BNE     15$             ;;EXIT IF NOT
066444  105777  112510          TSTB    @$TKS           ;;IS A CHAR WAITING?
066450  100121                  BPL     15$             ;;IF NOT, EXIT
066452  117746  112504          MOVB    @$TKB,-(SP)     ;;YES
066456  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
066462  021627  000007          CMP     (SP),#7         ;;IS IT A CONTROL-G?
066466  001300                  BNE     2$              ;;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.
066470  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
066476  001674                  BEQ     2$              ;;BRANCH IF YES
066500  005726                  TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
066502  004737  066146          JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
066506  005077  112446          CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
066512  112737  000001  001151  MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

066520  104401  067344          TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
066524  104401  067351  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
066530  013746  000176          MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
```

```
066534  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
066536  104401  067362                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
066542  005046                  19$:     CLR     -(SP)           ;;CLEAR COUNTER
066544  005046                          CLR     -(SP)           ;;THE NEW SWR
066546  105777  112406          7$:     TSTB    @$TKS           ;;CHAR THERE?
066552  100375                          BPL     7$              ;;IF NOT TRY AGAIN

066554  117746  112402                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
066560  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

066564  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
066570  001015                          BNE     9$              ;;BRANCH IF NOT
066572  104401  067332                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
066576  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
066602  123727  001151  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
066610  001003                          BNE     8$              ;;BRANCH IF NO
066612  012777  000100  112340          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
066620  000137  064154          8$:     JMP     SHUT2           ;;CONTROL-C RESTART


066624  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
066630  001005                          BNE     10$             ;;BRANCH IF NOT
066632  104401  067337                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
066636  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
066642  000737                          BR      19$             ;;LET'S TRY IT AGAIN


066644  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
066650  001022                          BNE     16$             ;;BRANCH IF NO
066652  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
066656  001403                          BEQ     11$             ;;BRANCH IF YES
066660  016677  000002  112266          MOV     2(SP),@SWR      ;;SAVE NEW SWR
066666  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
066672  104401  001217          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
066676  123727  001151  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
066704  001003                          BNE     15$             ;;BRANCH IF NOT
066706  012777  000100  112244          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
066714  000002                  15$:    RTI                     ;;RETURN
066716  004737  065220          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
066722  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
066726  002420                          BLT     18$             ;;BRANCH IF YES
066730  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
066734  003015                          BGT     18$             ;;BRANCH IF YES
066736  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
066742  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
066746  001403                          BEQ     17$             ;;BRANCH IF YES
066750  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
066752  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
066754  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
066756  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
066762  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
066766  000667                          BR      7$              ;;GET THE NEXT ONE
066770  104401  001216          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
066774  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
                                        ;

066776  011646                  $RDCHR: MOV     (SP),-(SP)          ;;PUSH DOWN THE PC AND
067000  016666  000004  000002          MOV     4(SP),2(SP)         ;;THE PS
067006  005066  000004                  CLR     4(SP)               ;;GET READY FOR A CHARACTER
067012  005046                          CLR     -(SP)               ;;PUT NEW PS ON STACK
067014  012746  067022                  MOV     #64$,-(SP)          ;;PUT NEW PC ON  STACK
067020  000002                          RTI                         ;;POP NEW PC AND PS
067022                          64$:
067022  005737  066136          1$:     TST     $TKCNT              ;;WAIT ON A CHARACTER
067026  001775                          BEQ     1$
067030  005337  066136                  DEC     $TKCNT              ;;DECREMENT THE COUNTER
067034  117766  177102  000004          MOVB    @$TKQOUT,4(SP)      ;;GET ONE CHARACTER
067042  005237  066142                  INC     $TKQOUT             ;;UPDATE THE POINTER
067046  023727  066142  066145          CMP     $TKQOUT,#$TKQEND    ;;DID IT GO OFF OF THE END?
067054  001003                          BNE     2$                  ;;BRANCH IF NO
067056  012737  066144  066142          MOV     #$TKQSRT,$TKQOUT    ;;RESET THE POINTER
067064  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

067066  010346                  $RDLIN: MOV     R3,-(SP)            ;;SAVE R3
067070  005046                          CLR     -(SP)               ;;CLEAR THE RUBOUT KEY
067072  012703  067322          1$:     MOV     #$TTYIN,R3          ;;GET ADDRESS
067076  022703  067332          2$:     CMP     #$TTYIN+8.,R3       ;;BUFFER FULL?
067102  101456                          BLOS    4$                  ;;BR IF YES
067104  104411                          RDCHR                       ;;GO READ ONE CHARACTER FROM THE TTY
067106  112613                          MOVB    (SP)+,(R3)          ;;GET CHARACTER
067110  122713  000177          10$:    CMPB    #177,(R3)           ;;IS IT A RUBOUT
067114  001022                          BNE     5$                  ;;BR IF NO
067116  005716                          TST     (SP)                ;;IS THIS THE FIRST RUBOUT?
067120  001007                          BNE     6$                  ;;BR IF NO
067122  112737  000134  067320          MOVB    #'\,9$              ;;TYPE A BACK SLASH
067130  104401  067320                  TYPE    ,9$
067134  012716  177777                  MOV     #-1,(SP)            ;;SET THE RUBOUT KEY
067140  005303                  6$:     DEC     R3                  ;;BACKUP BY ONE
067142  020327  067322                  CMP     R3,#$TTYIN          ;;STACK EMPTY?
067146  103434                          BLO     4$                  ;;BR IF YES
067150  111337  067320                  MOVB    (R3),9$             ;;SETUP TO TYPEOUT THE DELETED CHAR.
067154  104401  067320                  TYPE    ,9$                 ;;GO TYPE
067160  000746                          BR      2$                  ;;GO READ ANOTHER CHAR.
067162  005716                  5$:     TST     (SP)                ;;RUBOUT KEY SET?
067164  001406                          BEQ     7$                  ;;BR IF NO
067166  112737  000134  067320          MOVB    #'\,9$              ;;TYPE A BACK SLASH
067174  104401  067320                  TYPE    ,9$
067200  005016                          CLR     (SP)                ;;CLEAR THE RUBOUT KEY
067202  122713  000025          7$:     CMPB    #25,(R3)            ;;IS CHARACTER A CTRL J?
```

```
067206  001003                          BNE     8$              ;;BR IF NO
067210  104401  067337                  TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U'
067214  000726                          BR      1$              ;;GO START OVER
067216  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A '^R'?
067222  001011                          BNE     3$              ;;BRANCH IF NO
067224  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
067226  104401  001217                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
067232  104401  067322                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
067236  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
067240  104401  001216          4$:     TYPE    ,$QUES          ;;TYPE A '?'
067244  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
067246  111337  067320          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
067252  104401  0673??                  TYPE    ,9$
067256  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
067262  001305                          BNE     2$              ;;LOOP IF NOT RETURN
067264  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
067270  104401  001220                  TYPE    ,$LF            ;;TYPE A LINE FEED
067274  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
067276  012603                          MOV     (SP)+,R3        ;;RESTORE R3
067300  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
067302  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
067310  012766  067322  000004          MOV     #$TTYIN,4(SP)
067316  000002                          RTI                     ;;RETURN
067320     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
067321     000                  .BYTE   0                       ;;TERMINATOR
067322                  $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
067332     136   103   015      $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
067337     136   125   015      $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U'
067344     136   107   015      $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL 'G'
067351     015   012   123      $MSWR:  .ASCIZ  <15><12>/SWR = /
067362     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

067374  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
067376  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
067404  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
067406  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
067410  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
067412  104412          1$:     RDLIN                           ;;READ AN ASCIZ LINE
067414  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
067416  005001                          CLR     R1              ;;CLEAR DATA WORD
067420  005002                          CLR     R2
067422  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
067424  001412                          BEQ     3$              ;;IF ZERO GET OUT
067426  006301                          ASL     R1              ;;*2
067430  006102                          ROL     R2
067432  006301                          ASL     R1              ;;*4
067434  006102                          ROL     R2
```

```
067436  006301                          ASL     R1              ;;*8
067440  006102                          ROL     R2
067442  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
067446  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
067450  000764                          BR      2$              ;;LOOP
067452  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
067454  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
067460  010237  067474                  MOV     R2,$HIOCT
067464  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
067466  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
067470  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
067472  000002                          RTI                     ;;RETURN
067474  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.

067476  016646  000002          $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
067502  042716  000020                  BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
067506  012746  067514                  MOV     #1$,-(SP)       ;; T-BIT TRAPS
067512  000002                          RTI                     ;;SET THE NEW STATUS
067514  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
067516  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
067522  005740                          TST     -(R0)           ;;BACKUP BY 2
067524  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
067526  006300                          ASL     R0              ;;POSITION FOR INDEXING
067530  016000  067550                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
067534  000200                          RTS     R0              ;;GO TO ROUTINE


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

067536  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
067540  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
067546  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
067550  067536                  $TRPAD: .WORD   $TRAP2
067552  065006                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
067554  064604                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
067556  064560                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
067560  064620                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
067562  064334                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
067564  064260                          $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

067566  066524                          $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING
```

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

                                        ;;**********************************************************************
                                        ;POWER DOWN ROUTINE
                                        $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
                                                MOV     #340,@#PWRVEC+2 ;;PRIO:7
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                                MOV     R4,-(SP)        ;;PUSH R4 ON STACK
                                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
                                                MOV     SP,$SAVR6       ;;SAVE SP
                                                MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
                                                HALT
                                                BR      .-2             ;;HANG UP

                                        ;;**********************************************************************
                                        ;POWER UP ROUTINE
                                        $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
                                                MOV     $SAVR6,SP       ;;GET SP
                                                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
                                        1$:     INC     $SAVR6          ;;WAIT FOR THE INC
                                                BNE     1$              ;;OF  WORD
                                                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
                                                MOV     (SP)+,R5        ;;POP STACK INTO R5
                                                MOV     (SP)+,R4        ;;POP STACK INTO R4
                                                MOV     (SP)+,R3        ;;POP STACK INTO R3
                                                MOV     (SP)+,R2        ;;POP STACK INTO R2
                                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                                MOV     (SP)+,R0        ;;POP STACK INTO R0
                                                MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
                                                MOV     #340,@#PWRVEC+2 ;;PRIO:7
                                                TYPE                    ;REPORT THE POWER FAILURE
                                        $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
                                                RTI
                                        $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
                                                BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
                                        $SAVR6: 0                       ;;PUT THE SP HERE
                                        $POWER: .ASCIZ  <15><12>'POWER'
                                                .EVEN
    12                                  .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;**********************************************************************
                                        $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
                                        $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
                                                BR      $ATYC
                                        $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
                                        $ATYC:
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
070010  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
070012  105737  070224                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
070016  001450                          BEQ     5$              ;;IF NOT:  BR
070020  122737  000001  001242          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
070026  001031                          BNE     3$              ;;IF NOT:  BR
070030  132737  000100  001243          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
070036  001425                          BEQ     3$              ;;IF NOT:  BR
070040  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
070044  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
070052  005737  001222          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
070056  001375                          BNE     1$              ;;IF NOT:  WAIT
070060  010037  001236                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
070064  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
070066  001376                          BNE     2$
070070  163700  001236                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
070074  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
070076  010037  001240                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
070102  012737  000004  001222          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
070110  000413                          BR      5$
070112  017637  000004  070136  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
070120  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
070126  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
070132  004737  065006                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
070136  000000                  4$:     .WORD   0
070140                          5$:
070140  105737  070226          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
070144  001416                          BEQ     12$             ;;IF NOT:  BR
070146  005737  001242                  TST     $ENV            ;;RUNNING UNDER APT?
070152  001413                          BEQ     12$             ;;IF NOT:  BR
070154  005737  001222          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
070160  001375                          BNE     11$             ;;IF NOT:  WAIT
070162  017637  000004  001224          MOV     @4(SP),$FATAL   ;;GET ERROR #
070170  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
070176  005237  001222                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
070202  105037  070226          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
070206  105037  070225                  CLRB    $LFLG           ;;CLEAR LOG FLAG
070212  105037  070224                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
070216  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
070220  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
070222  000207                          RTS     PC              ;;RETURN
070224     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
070225     000          $LFLG:  .BYTE   0               ;;LOG FLAG
070226     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
                                .EVEN
             000200     APTSIZE = 200
             000001     APTENV  = 001
             000100     APTSPOOL= 100
             000040     APTCSUP = 040
```

```
                                        .SBTTL  CONSOLE MESSAGES

070230  200  106  101  SCTMSG: .ASCII  <CRLF>@FAILED TO RECOVER THE BAD SECTOR FILE (DEC 144)@
070310  200  117  116         .ASCIZ  <CRLF>@ON THIS DRIVE@
070327  075  000       EQUALS: .ASCIZ  @=@
070331  101  114  114  ALL:    .ASCIZ  @ALL@<CRLF>
070336  040  077  040  QUES:   .ASCIZ  @ ? @
070342  054  040  000  COMMA:  .ASCIZ  @, @
070345  200  124  117  MSHELP: .ASCII  <CRLF>@TO  ENSURE THAT NO  BAD HEADERS ARE LEFT  ON THE DISK@
070433  200  120  101         .ASCII  <CRLF>@PACK,  THIS PROGRAM SHOULD BE HALTED BY TYPING A (^C)@
070521  200  103  117         .ASCII  <CRLF>@CONTROL C.  AS A RESULT,  THE PROGRAM WILL  BE HALTED@
070607  200  127  110         .ASCII  <CRLF>@WHEN THE DRIVE UNDER TEST HAS COMPLETED TESTING.@<CRLF>
070671  200  124  131         .ASCIZ  <CRLF>@TYPE HELP TEXT (Y/N) ? @
070722                  UBUSQST:
070722  200  103  110         .ASCIZ  <CRLF>@CHANGE ADDRESSES (Y/N) ? @
070755  200  125  123  CNSL00: .ASCIZ  <CRLF>@USE SAME DEVICES (Y/N) ? @
071010  200  102  125  CNSL01: .ASCIZ  <CRLF>@BUS ADDRESS @
071026  040  114  111  CNSL02: .ASCIZ  @ LIMITS - LO= 160000, HI= 17XXXX@<CRLF>
071070  126  105  103  CNSL03: .ASCIZ  @VECTOR ADDRESS @
071110  040  114  111  CNSL04: .ASCIZ  @ LIMITS - LO= 0, HI= 1000@<CRLF><LF>
071144  102  122  040  CNSL05: .ASCIZ  @BR LEVEL @
071156  040  114  111  CNSL06: .ASCIZ  @ LIMITS - LO= 0, HI= 7@<CRLF><LF>
071207  200               CNSL07: .ASCII  <CRLF>
071210  200  124  131         .ASCII  <CRLF>@TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S)@
071275  200  101  116         .ASCIZ  <CRLF>@AND TERMINATE INPUT WITH A CARRIAGE RETURN.@
071352  200               CNSL08: .ASCII  <CRLF>
071353  040  077  111  CNSL09: .ASCIZ  @ ?ILLEGAL INPUT@<CRLF>
071374  200  104  122  MSDRVS: .ASCIZ  <CRLF>/DRIVE(S): /
071410  104  122  111  MSGDRV: .ASCIZ  /DRIVE/
071416  040  111  123  LODEV:  .ASCIZ  / IS LOAD DEVICE/
071436  040  116  117  NOTPRS: .ASCIZ  / NOT PRESENT/
071453  040  116  117  NOTAVL: .ASCIZ  / NOT AVAILABLE/
071472  040  117  106  OFFLIN: .ASCIZ  / OFF LINE/
071504  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', 'WCE', '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.

071516                          FNCDTB:                                 ;FUNCTION CODE TABLE

071516  020000                  .WORD   OPI                             ;NOP
```

```
     58 071520  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (2)
     59 071522  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEEK
     60 071524  130000                .WORD   ATA!OPI!IVC                     ;RECALIBRATE
     61 071526  020000                .WORD   OPI                             ;DRIVE CLEAR
     62 071530  030000                .WORD   OPI!IVC                         ;RELEASE
     63 071532  130000                .WORD   OPI!ATA!IVC                     ;OFFSET
     64 071534  130000                .WORD   OPI!ATA!IVC                     ;RETURN TO CENTERLINE
     65 071536  020000                .WORD   OPI                             ;READ IN PRESET
     66 071540  020000                .WORD   OPI                             ;PACK ACKNOWLEDGE
     67 071542  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (24)
     68 071544  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (26)
     69 071546  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEARCH
     70 071550  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (32)
     71 071552  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (34)
     72 071554  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (36)
     73 071556  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (40)
     74 071560  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (42)
     75 071562  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (44)
     76 071564  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (46)
     77 071566  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
     78 071570  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK HEADER AND DATA
     79 071572  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (54)
     80 071574  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (56)
     81 071576  037200                .WORD   OPI!IVC!WLE!IAE!AOE!HCE         ;WRITE DATA
     82 071600  037000                .WORD   OPI!IVC!WLE!IAE!AOE             ;WRITE HEADER AND DATA
     83 071602  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (64)
     84 071604  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (66)
     85 071606  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ DATA
     86 071610  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ HEADER AND DATA
     87 071612  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (74)
     88 071614  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (76)
     89
```

SEQ 0276

```
                                .SBTTL  ATTENTION (ATA) TABLE

071616      001         ATNTBL: .BYTE   1.
071617      002                 .BYTE   2.
071620      004                 .BYTE   4.
071621      010                 .BYTE   8.
071622      020                 .BYTE   16.
071623      040                 .BYTE   32.
071624      100                 .BYTE   64.
071625      200                 .BYTE   128.
```

```
                                .SBTTL  DATA PATTERN TABLE

071626                          RGDTPT:
071626                          MIXED:
071626  000000                          .WORD   0.
071630  000001                          .WORD   1.
071632  000003                          .WORD   3.
071634  000007                          .WORD   7.
071636  000017                          .WORD   15.
071640  000037                          .WORD   31.
071642  000077                          .WORD   63.
071644  000177                          .WORD   127.
071646  000377                          .WORD   255.
071650  000777                          .WORD   511.
071652  001777                          .WORD   1023.
071654  003777                          .WORD   2047.
071656  007777                          .WORD   4095.
071660  017777                          .WORD   8191.
071662  037777                          .WORD   16383.
071664  077777                          .WORD   32767.
071666  177777                  ONES:   .WORD   65535.
071670  177777                          .WORD   65535.
071672  077777                          .WORD   32767.
071674  037777                          .WORD   16383.
071676  017777                          .WORD   8191.
071700  007777                          .WORD   4095.
071702  003777                          .WORD   2047.
071704  001777                          .WORD   1023.
071706  000777                          .WORD   511.
071710  000377                          .WORD   255.
071712  000177                          .WORD   127.
071714  000077                          .WORD   63.
071716  000037                          .WORD   31.
071720  000017                          .WORD   15.
071722  000007                          .WORD   7.
071724  000003                          .WORD   3.
071726  000001                          .WORD   1.
071730  000000                  ZEROS:  .WORD   0.
071732  000000                          .WORD   0.
071734  000001                          .WORD   1.
071736  000002                          .WORD   2.
071740  000004                          .WORD   4.
071742  000010                          .WORD   8.
071744  000020                          .WORD   16.
071746  000040                          .WORD   32.
071750  000100                          .WORD   64.
071752  000200                          .WORD   128.
071754  000400                          .WORD   256.
071756  001000                          .WORD   512.
071760  002000                          .WORD   1024.
071762  004000                          .WORD   2048.
071764  010000                          .WORD   4096.
071766  020000                          .WORD   8192.
071770  040000                          .WORD   16384.
071772  100000                          .WORD   32768.
071774  100000                          .WORD   32768.
071776  040000                          .WORD   16384.
```

```
 58 072000  020000                    .WORD   8192.
 59 072002  010000                    .WORD   4096.
 60 072004  004000                    .WORD   2048.
 61 072006  002000                    .WORD   1024.
 62 072010  001000                    .WORD   512.
 63 072012  000400                    .WORD   256.
 64 072014  000200                    .WORD   128.
 65 072016  000100                    .WORD   64.
 66 072020  000040                    .WORD   32.
 67 072022  000020                    .WORD   16.
 68 072024  000010                    .WORD   8.
 69 072026  000004                    .WORD   4.
 70 072030  000002                    .WORD   2.
 71 072032  000001                    .WORD   1.
 72 072034  000000                    .WORD   0.
 73 072036  177777                    .WORD   65535.
 74 072040  177776                    .WORD   65534.
 75 072042  177774                    .WORD   65532.
 76 072044  177770                    .WORD   65528.
 77 072046  177760                    .WORD   65520.
 78 072050  177740                    .WORD   65504.
 79 072052  177700                    .WORD   65472.
 80 072054  177600                    .WORD   65408.
 81 072056  177400                    .WORD   65280.
 82 072060  177000                    .WORD   65024.
 83 072062  176000                    .WORD   64512.
 84 072064  174000                    .WORD   63488.
 85 072066  170000                    .WORD   61440.
 86 072070  160000                    .WORD   57344.
 87 072072  140000                    .WORD   49152.
 88 072074  100000                    .WORD   32768.
 89 072076  000000                    .WORD   0.
 90 072100  000000                    .WORD   0.
 91 072102  100000                    .WORD   32768.
 92 072104  140000                    .WORD   49152.
 93 072106  160000                    .WORD   57344.
 94 072110  170000                    .WORD   61440.
 95 072112  174000                    .WORD   63488.
 96 072114  176000                    .WORD   64512.
 97 072116  177000                    .WORD   65024.
 98 072120  177400                    .WORD   65280.
 99 072122  177600                    .WORD   65408.
100 072124  177700                    .WORD   65472.
101 072126  177740                    .WORD   65504.
102 072130  177760                    .WORD   65520.
103 072132  177770                    .WORD   65528.
104 072134  177774                    .WORD   65532.
105 072136  177776                    .WORD   65534.
106 072140  177777                    .WORD   65535.
107 072142  125252          EARLY:    .WORD   43690.
108 072144  152525                    .WORD   43690./2
109 072146  125252                    .WORD   43690.
110 072150  177777                    .WORD   65535.
111 072152  177776                    .WORD   65534.
112 072154  177775                    .WORD   65533.
113 072156  177773                    .WORD   65531.
114 072160  177767                    .WORD   65527.
```

```
    115 072162  177757                          .WORD   65519.
    116 072164  177737                          .WORD   65503.
    117 072166  177677                          .WORD   65471.
    118 072170  177577                          .WORD   65407.
    119 072172  177377                          .WORD   65279.
    120 072174  176777                          .WORD   65023.
    121 072176  175777                          .WORD   64511.
    122 072200  173777                          .WORD   63487.
    123 072202  167777                          .WORD   61439.
    124 072204  157777                          .WORD   57343.
    125 072206  137777                          .WORD   49151.
    126 072210  077777                          .WORD   32767.
    127 072212  077777                          .WORD   32767.
    128 072214  137777                          .WORD   49151.
    129 072216  157777                          .WORD   57343.
    130 072220  167777                          .WORD   61439.
    131 072222  173777                          .WORD   63487.
    132 072224  175777                          .WORD   64511.
    133 072226  176777                          .WORD   65023.
    134 072230  177377                          .WORD   65279.
    135 072232  177577                          .WORD   65407.
    136 072234  177677                          .WORD   65471.
    137 072236  177737                          .WORD   65503.
    138 072240  177757                          .WORD   65519.
    139 072242  177767                          .WORD   65527.
    140 072244  177773                          .WORD   65531.
    141 072246  177775                          .WORD   65533.
    142 072250  177776                          .WORD   65534.
    143 072252  177777                          .WORD   65535.
    144 072254                          ENRGDT:
    145
```

```
                                        .SBTTL  ERROR MESSAGE TABLE

072254  076650  000000          EMT1:   .WORD   EMS1,0
072260  076717  076734  000000  EMT2:   .WORD   EMS2,EMS3,0
072266  076717  076777  000000  EMT3:   .WORD   EMS2,EMS4,0
072274  077042  077072  000000  EMT4:   .WORD   EMS5,EMS6,0
072302  077042  077204  000000  EMT5:   .WORD   EMS5,EMS10,0
072310  103644  101063  000000  EMT6:   .WORD   EMS167,EMS64,0
072316  101631  103671  000000  EMT7:   .WORD   EMS110,EMS170,0
072324  077137  000000          EMT10:  .WORD   EMS7,0
072330  077204  000000          EMT11:  .WORD   EMS10,0
072334  077246  077257  000000  EMT12:  .WORD   EMS11,EMS12,0
072342  077320  077331  077342  EMT13:  .WORD   EMS13,EMS14,EMS15,EMS16,0
072354  077414  101063  000000  EMT14:  .WORD   EMS17,EMS64,0
072362  077246  077477  000000  EMT15:  .WORD   EMS11,EMS21,0
072370  077246  077522  077651  EMT16:  .WORD   EMS11,EMS22,EMS27,0
072400  077246  077536  077662  EMT17:  .WORD   EMS11,EMS23,EMS30,0
072410  077246  077564  077662  EMT20:  .WORD   EMS11,EMS24,EMS30,0
072420  077246  077613  077651  EMT21:  .WORD   EMS11,EMS25,EMS27,0
072430  077246  077630  077651  EMT22:  .WORD   EMS11,EMS26,EMS27,0
072440  077246  077672  077662  EMT23:  .WORD   EMS11,EMS31,EMS30,0
072450  077246  077721  077662  EMT24:  .WORD   EMS11,EMS32,EMS30,0
072460  077246  077750  077662  EMT25:  .WORD   EMS11,EMS33,EMS30,0
072470  077246  077776  077662  EMT26:  .WORD   EMS11,EMS34,EMS30,0
072500  077246  100047  077662  EMT27:  .WORD   EMS11,EMS35,EMS30,0
072510  077246  100076  077662  EMT30:  .WORD   EMS11,EMS36,EMS30,0
072520  077246  100125  077662  EMT31:  .WORD   EMS11,EMS37,EMS30,0
072530  077246  100153  077662  EMT32:  .WORD   EMS11,EMS40,EMS30,0
072540  077246  100202  077662  EMT33:  .WORD   EMS11,EMS41,EMS30,0
072550  077246  100230  077662  EMT34:  .WORD   EMS11,EMS42,EMS30,0
072560  077246  100257  077662  EMT35:  .WORD   EMS11,EMS43,EMS30,0
072570  077246  100306  077662  EMT36:  .WORD   EMS11,EMS44,EMS30,0
072600  077246  100361  077662  EMT37:  .WORD   EMS11,EMS45,EMS30,0
072610  101147  077454  000000  EMT40:  .WORD   EMS66,EMS20,0
072616  101335  103046  101343  EMT41:  .WORD   EMS75,EMS141,EMS76,0
072626  103137  103147  101271  EMT42:  .WORD   EMS144,EMS145,EMS72,EMS76,0
072640  100453  100567  101343  EMT43:  .WORD   EMS47,EMS53,EMS76,0
072650  101374  100567  101343  EMT44:  .WORD   EMS77,EMS53,EMS76,0
072660  101422  100567  101343  EMT45:  .WORD   EMS100,EMS53,EMS76,0
072670  101450  100567  101343  EMT46:  .WORD   EMS101,EMS53,EMS76,0
072700  101101  101063  000000  EMT47:  .WORD   EMS65,EMS64,0
072706  077320  077342  101035  EMT50:  .WORD   EMS13,EMS15,EMS63,0
072716  077246  100424  077662  EMT51:  .WORD   EMS11,EMS46,EMS30,EMS67,0
072730  100453  100567  101217  EMT52:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS140,EMS141,0
072746  100453  100567  101217  EMT53:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS141,EMS164,0
072764  100502  100567  101217  EMT54:  .WORD   EMS50,EMS53,EMS67,0
072774  103074  103111  100567  EMT55:  .WORD   EMS142,EMS143,EMS53,EMS67,0
073006  100531  101271  101217  EMT56:  .WORD   EMS51,EMS72,EMS67,EMS115,EMS50,EMS70,0
073024  103644  101301  101217  EMT57:  .WORD   EMS167,EMS73,EMS67,0
073034  100652  101217  102031  EMT60:  .WORD   EMS56,EMS67,EMS115,EMS150,EMS152,EMS70,0
073052  101256  100652  101217  EMT61:  .WORD   EMS71,EMS56,EMS67,EMS115,EMS150,EMS152,EMS72,0
073072  103025  101217  102031  EMT62:  .WORD   EMS140,EMS67,EMS115,EMS47,EMS70,0
073106  000000                  EMT63:  .WORD
073110  103232  103274  101244  EMT64:  .WORD   EMS150,EMS152,EMS70,EMS115,EMS56,EMS73,EMS67,0
073130  100557  104271  104452  EMT65:  .WORD   EMS52,EMS205,EMS214,EMS206,EMS115,EMS51,EMS72,0
073150  103603  101525  101271  EMT66:  .WORD   EMS165,EMS103,EMS72,EMS124,0
073162  103603  101525  101271  EMT67:  .WORD   EMS165,EMS103,EMS72,EMS171,0
```

```
 58 073174  100424  077454  103476  EMT70:  .WORD   EMS46,EMS20,EMS163,0
 59 073204  101256  101450  103476  EMT71:  .WORD   EMS71,EMS101,EMS163,0
 60 073214  100453  101271  103476  EMT72:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS140,EMS141,0
 61 073232  100453  101271  103476  EMT73:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS141,EMS72,0
 62 073250  100652  100567  103476  EMT74:  .WORD   EMS56,EMS53,EMS163,0
 63 073260  101256  100652  103476  EMT75:  .WORD   EMS71,EMS56,EMS163,EMS115,EMS150,EMS152,EMS72,0
 64 073300  100502  100567  103476  EMT76:  .WORD   EMS50,EMS53,EMS163,0
 65 073310  103074  103111  100567  EMT77:  .WORD   EMS142,EMS143,EMS53,EMS163,0
 66 073322  101335  103046  103476  EMT100: .WORD   EMS75,EMS141,EMS163,EMS115,EMS47,EMS70,0
 67 073340  101335  103232  103476  EMT101: .WORD   EMS75,EMS150,EMS163,EMS115,EMS56,EMS73,0
 68 073356  103644  101301  103476  EMT102: .WORD   EMS167,EMS73,EMS163,0
 69 073366  103215  103250  101316  EMT103: .WORD   EMS147,EMS151,EMS74,EMS163,0
 70 073400  100531  101316  103476  EMT104: .WORD   EMS51,EMS74,EMS163,EMS115,EMS50,EMS70,0
 71 073416  103603  101422  103476  EMT105: .WORD   EMS165,EMS100,EMS163,0
 72 073426  103603  101374  103476  EMT106: .WORD   EMS165,EMS77,EMS163,0
 73 073436  103137  103147  100567  EMT107: .WORD   EMS144,EMS145,EMS53,EMS163,EMS115,EMS143,EMS70,0
 74 073456  101631  101671  102034  EMT110: .WORD   EMS110,EMS112,EMS116,EMS111,0
 75 073470  101755  076777  000000  EMT111: .WORD   EMS113,EMS4,0
 76 073476  101755  076734  000000  EMT112: .WORD   EMS113,EMS3,0
 77 073504  101631  102031  102034  EMT113: .WORD   EMS110,EMS115,EMS116,EMS117,EMS114,0
 78 073520  101755  102106  000000  EMT114: .WORD   EMS113,EMS120,0
 79 073526  102127  102567  102613  EMT115: .WORD   EMS121,EMS132,EMS133,0
 80 073536  102172  102567  102613  EMT116: .WORD   EMS122,EMS132,EMS133,0
 81 073546  102227  102567  102613  EMT117: .WORD   EMS123,EMS132,EMS133,0
 82 073556  102272  102567  102613  EMT120: .WORD   EMS124,EMS132,EMS133,
 83 073566  102324  102567  102613  EMT121: .WORD   EMS125,EMS132,EMS133,0
 84 073576  102367  102567  102613  EMT122: .WORD   EMS126,EMS132,EMS133,0
 85 073606  102770  102567  102613  EMT123: .WORD   EMS137,EMS132,EMS133,0
 86 073616  102471  102567  102613  EMT124: .WORD   EMS130,EMS132,EMS133,0
 87 073626  102527  102567  102613  EMT125: .WORD   EMS131,EMS132,EMS133,0
 88 073636  102127  102567  102636  EMT126: .WORD   EMS121,EMS132,EMS134,EMS123,0
 89 073650  102172  102567  102636  EMT127: .WORD   EMS122,EMS132,EMS134,EMS123,0
 90 073662  102227  102567  102636  EMT130: .WORD   EMS123,EMS132,EMS134,EMS123,0
 91 073674  102272  102567  102636  EMT131: .WORD   EMS124,EMS132,EMS134,EMS123,0
 92 073706  102324  102567  102636  EMT132: .WORD   EMS125,EMS132,EMS134,EMS123,0
 93 073720  102367  102567  102636  EMT133: .WORD   EMS126,EMS132,EMS134,EMS123,0
 94 073732  102770  102567  102636  EMT134: .WORD   EMS137,EMS132,EMS134,EMS123,0
 95 073744  102471  102567  102636  EMT135: .WORD   EMS130,EMS132,EMS134,EMS123,0
 96 073756  102527  102567  102636  EMT136: .WORD   EMS131,EMS132,EMS134,EMS123,0
 97 073770  102127  102567  102700  EMT137: .WORD   EMS121,EMS132,EMS135,0
 98 074000  102227  102567  102700  EMT140: .WORD   EMS123,EMS132,EMS135,0
 99 074010  102127  102567  102743  EMT141: .WORD   EMS121,EMS132,EMS136,0
100 074020  102770  102567  102743  EMT142: .WORD   EMS137,EMS132,EMS136,0
101 074030  102272  102567  102743  EMT143: .WORD   EMS124,EMS132,EMS136,0
102 074040  102324  102567  102743  EMT144: .WORD   EMS125,EMS132,EMS136,0
103 074050  102367  102567  102743  EMT145: .WORD   EMS126,EMS132,EMS136,0
104 074060  102527  102567  102743  EMT146: .WORD   EMS131,EMS132,EMS136,0
105 074070  103720  102567  102743  EMT147: .WORD   EMS171,EMS132,EMS136,0
106 074100  102471  102567  102743  EMT150: .WORD   EMS130,EMS132,EMS136,0
107 074110  103025  102031  103046  EMT151: .WORD   EMS140,EMS115,EMS141,EMS70,0
108 074122  103074  102031  103111  EMT152: .WORD   EMS142,EMS115,EMS143,EMS72,0
109 074134  103137  103147  103176  EMT153: .WORD   EMS144,EMS145,EMS146,EMS115,EMS143,EMS72,0
110 074152  103137  103147  077454  EMT154: .WORD   EMS144,EMS145,EMS20,EMS115,EMS143,EMS70,0
111 074170  103232  103274  103342  EMT155: .WORD   EMS150,EMS152,EMS154,EMS153,0
112 074202  103215  103250  103342  EMT156: .WORD   EMS147,EMS151,EMS154,EMS155,0
113 074214  103215  103250  103376  EMT157: .WORD   EMS147,EMS151,EMS156,EMS157,0
114 074226  103446  103376  103431  EMT160: .WORD   EMS161,EMS156,EMS160,0
```

```
    115 074236  077613  077651  103376  EMT161: .WORD   EMS25,EMS27,EMS156,EMS160,0
    116 074250  077630  077651  103376  EMT162: .WORD   EMS26,EMS27,EMS156,EMS160,0
    117 074262  100453  103476  103025  EMT163: .WORD   EMS47,EMS163,EMS140,0
    118 074272  100453  077454  000000  EMT164: .WORD   EMS47,EMS20,0
    119 074300  100652  077454  000000  EMT165: .WORD   EMS56,EMS20,0
    120 074306  100424  077454  000000  EMT166: .WORD   EMS46,EMS20,0
    121 074314  104772  077454  000000  EMT167: .WORD   EMS224,EMS20,0
    122 074322  103644  103342  103617  EMT170: .WORD   EMS167,EMS154,EMS166,0
    123 074332  103644  103342  103414  EMT171: .WORD   EMS167,EMS154,EMS157,0
    124 074342  103644  103342  103356  EMT172: .WORD   EMS167,EMS154,EMS155,0
    125 074352  103720  102567  102613  EMT173: .WORD   EMS171,EMS132,EMS133,0
    126 074362  103720  102567  102636  EMT174: .WORD   EMS171,EMS132,EMS134,EMS123,0
    127 074374  101755  104073  000000  EMT175: .WORD   EMS113,EMS177,0
    128 074402  104115  104132  000000  EMT176: .WORD   EMS200,EMS201,0
    129 074410  104230  103046  077662  EMT177: .WORD   EMS203,EMS141,EMS30,EMS202,0
    130 074422  104230  100531  077662  EMT200: .WORD   EMS203,EMS51,EMS30,EMS202,0
    131 074434  104230  104243  077662  EMT201: .WORD   EMS203,EMS204,EMS30,EMS202,0
    132 074446  104230  100502  077662  EMT202: .WORD   EMS203,EMS50,EMS30,EMS202,0
    133 074460  104230  103111  077662  EMT203: .WORD   EMS203,EMS143,EMS30,EMS202,0
    134 074472  103232  101316  104200  EMT204: .WORD   EMS150,EMS74,EMS202,EMS115,EMS152,EMS72,0
    135 074510  100502  101525  101271  EMT205: .WORD   EMS50,EMS103,EMS72,0
    136 074520  101534  101301  104200  EMT206: .WORD   EMS104,EMS73,EMS202,0
    137 074530  101335  103046  102031  EMT207: .WORD   EMS75,EMS141,EMS115,EMS140,0
    138 074542  101335  103232  000000  EMT210: .WORD   EMS75,EMS150,0
    139 074550  100531  101271  102031  EMT211: .WORD   EMS51,EMS72,EMS115,EMS50,EMS70,0
    140 074564  103074  100567  102031  EMT212: .WORD   EMS142,EMS53,EMS115,EMS143,EMS72,0
    141 074600  100502  101525  100567  EMT213: .WORD   EMS50,EMS103,EMS53,0
    142 074610  100557  104271  103617  EMT214: .WORD   EMS52,EMS205,EMS166,EMS206,EMS115,EMS51,EMS72,0
    143 074630  100557  102064  100652  EMT215: .WORD   EMS52,EMS117,EMS56,EMS163,0
    144 074642  100652  077662  101063  EMT216: .WORD   EMS56,EMS30,EMS64,0
    145 074652  100424  077662  101063  EMT217: .WORD   EMS46,EMS30,EMS64,0
    146 074662  077672  077662  101063  EMT220: .WORD   EMS31,EMS30,EMS64,0
    147 074672  100453  0?'662  101063  EMT221: .WORD   EMS47,EMS30,EMS64,0
    148 074702  100557  102064  101374  EMT222: .WORD   EMS52,EMS117,EMS77,0
    149 074712  100557  102064  101035  EMT223: .WORD   EMS52,EMS117,EMS63,0
    150 074722  000000                  EMT224: .WORD
    151 074724  000000                  EMT225: .WORD
    152 074726  000000                  EMT226: .WORD
    153 074730  000000                  EMT227: .WORD
    154 074732  000000                  EMT230: .WORD
    155 074734  000000                  EMT231: .WORD
    156 074736  000000                  EMT232: .WORD
    157 074740  000000                  EMT233: .WORD
    158 074742  000000                  EMT234: .WORD
    159 074744  000000                  EMT235: .WORD
    160 074746  000000                  EMT236: .WORD
    161 074750  000000                  EMT237: .WORD
    162 074752  000000                  EMT240: .WORD
    163 074754  000000                  EMT241: .WORD
    164 074756  000000                  EMT242: .WORD
    165 074760  000000                  EMT243: .WORD
    166 074762  000000                  EMT244: .WORD
    167 074764  000000                  EMT245: .WORD
    168 074766  103644  102567  104330  EMT246: .WORD   EMS167,EMS132,EMS207,0
    169 074776  103644  102567  104355  EMT247: .WORD   EMS167,EMS132,EMS210,EMS125,0
    170 075010  103644  104366  104355  EMT250: .WORD   EMS167,EMS211,EMS210,EMS207,EMS206,0
    171 075024  104404  104427  000000  EMT251: .WORD   EMS212,EMS213,0
```

```
    172 075032  103603  104404  000000  EMT252: .WORD   EMS165,EMS212,0
    173 075040  103215  103250  103376  EMT253: .WORD   EMS147,EMS151,EMS156,EMS210,EMS26,EMS27,0
    174 075056  104230  101450  077662  EMT254: .WORD   EMS203,EMS101,EMS30,0
    175 075066  104230  103644  077662  EMT255: .WORD   EMS203,EMS167,EMS30,0
    176 075076  104230  101422  077662  EMT256: .WORD   EMS203,EMS100,EMS30,0
    177 075106  077246  100424  077662  EMT257: .WORD   EMS11,EMS46,EMS30,EMS102,0
    178 075120  100557  102064  100652  EMT260: .WORD   EMS52,EMS117,EMS56,EMS102,0
    179 075132  100557  104271  104574  EMT261: .WORD   EMS52,EMS205,EMS220,EMS206,EMS115,EMS51,EMS72,0
    180 075152  101534  101301  104200  EMT262: .WORD   EMS104,EMS73,EMS202,0
    181 075162  100502  100567  101476  EMT263: .WORD   EMS50,EMS53,EMS102,0
    182 075172  100652  100567  101476  EMT264: .WORD   EMS56,EMS53,EMS102,EMS115,EMS150,EMS152,EMS70,0
    183 075212  101256  100652  101476  EMT265: .WORD   EMS71,EMS56,EMS102,EMS115,EMS150,EMS152,EMS72,0
    184 075232  103074  103111  100567  EMT266: .WORD   EMS142,EMS143,EMS53,EMS102,0
    185 075244  100502  103176  102031  EMT267: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,0
    186 075262  100453  100567  101476  EMT270: .WORD   EMS47,EMS53,EMS102,EMS115,EMS140,0
    187 075276  100453  100567  101476  EMT271: .WORD   EMS47,EMS53,EMS102,EMS115,EMS141,EMS72,0
    188 075314  101335  103046  101476  EMT272: .WORD   EMS75,EMS141,EMS102,EMS115,EMS47,EMS73,0
    189 075332  100531  101316  101476  EMT273: .WORD   EMS51,EMS74,EMS102,EMS115,EMS50,EMS70,0
    190 075350  101035  100567  100727  EMT274: .WORD   EMS63,EMS53,EMS57,EMS115,EMS41,EMS146,0
    191 075366  102034  100567  100202  EMT275: .WORD   EMS116,EMS53,EMS41,EMS57,0
    192 075400  100453  100567  100727  EMT276: .WORD   EMS47,EMS53,EMS57,EMS115,EMS140,0
    193 075414  100453  100567  100727  EMT277: .WORD   EMS47,EMS53,EMS57,EMS115,EMS141,EMS72,0
    194 075432  100652  100567  100727  EMT300: .WORD   EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS70,0
    195 075452  101256  100652  100567  EMT301: .WORD   EMS71,EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS72,0
    196 075474  103603  101422  101525  EMT302: .WORD   EMS165,EMS100,EMS103,EMS57,0
    197 075506  103603  101450  101525  EMT303: .WORD   EMS165,EMS101,EMS103,EMS57,0
    198 075520  103603  101374  101525  EMT304: .WORD   EMS165,EMS77,EMS103,EMS57,0
    199 075532  100424  077662  101063  EMT305: .WORD   EMS46,EMS30,EMS64,EMS57,0
    200 075544  100502  100567  100727  EMT306: .WORD   EMS50,EMS53,EMS57,0
    201 075554  100502  103176  102031  EMT307: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,EMS57,0
    202 075574  103074  103111  100567  EMT310: .WORD   EMS142,EMS143,EMS53,EMS57,0
    203 075606  101534  101525  100567  EMT311: .WORD   EMS104,EMS103,EMS53,EMS57,0
    204 075620  101563  101525  100567  EMT312: .WORD   EMS105,EMS103,EMS53,EMS57,0
    205 075632  103137  103147  101525  EMT313: .WORD   EMS144,EMS145,EMS103,EMS57,EMS115,EMS143,EMS70,0
    206 075652  100230  101525  100567  EMT314: .WORD   EMS42,EMS103,EMS53,EMS57,0
    207 075664  077672  101525  100567  EMT315: .WORD   EMS31,EMS103,EMS53,EMS57,0
    208 075676  101256  077672  101525  EMT316: .WORD   EMS71,EMS31,EMS103,EMS57,0
    209 075710  100257  101525  100727  EMT317: .WORD   EMS43,EMS103,EMS57,0
    210 075720  100361  101525  100727  EMT320: .WORD   EMS45,EMS103,EMS57,0
    211 075730  100306  101525  100727  EMT321: .WORD   EMS44,EMS103,EMS57,0
    212 075740  101612  077454  000000  EMT322: .WORD   EMS106,EMS20,0
    213 075746  100076  101525  100727  EMT323: .WORD   EMS36,EMS103,EMS57,0
    214 075756  103773  100076  101525  EMT324: .WORD   EMS173,EMS36,EMS103,EMS57,0
    215 075770  103753  100076  101525  EMT325: .WORD   EMS172,EMS36,EMS103,EMS57,0
    216 076002  077320  104010  077342  EMT326: .WORD   EMS13,EMS174,EMS15,EMS35,EMS53,EMS175,0
    217 076020  103215  103250  101316  EMT327: .WORD   EMS147,EMS151,EMS74,EMS175,0
    218 076032  101147  100567  104016  EMT330: .WORD   EMS66,EMS53,EMS175,0
    219 076042  077750  101525  100567  EMT331: .WORD   EMS33,EMS103,EMS53,EMS175,0
    220 076054  100153  101525  100567  EMT332: .WORD   EMS40,EMS103,EMS53,EMS57,0
    221 076066  100531  101316  100727  EMT333: .WORD   EMS51,EMS74,EMS57,EMS115,EMS50,EMS70,0
    222 076104  101335  103046  100727  EMT334: .WORD   EMS75,EMS141,EMS57,EMS115,EMS47,EMS73,0
    223 076122  101335  103232  103274  EMT335: .WORD   EMS75,EMS150,EMS152,EMS57,EMS115,EMS56,EMS73,0
    224 076142  100755  100770  101017  EMT336: .WORD   EMS60,EMS61,EMS62,0
    225 076152  102034  102064  077776  EMT337: .WORD   EMS116,EMS117,EMS34,0
    226 076162  077776  100567  100601  EMT340: .WORD   EMS34,EMS53,EMS54,EMS111,0
    227 076174  100623  077776  000000  EMT341: .WORD   EMS55,EMS34,0
    228 076202  100557  102064  100652  EMT342: .WORD   EMS52,EMS117,EMS56,EMS57,0
```

```
                                                                    N 6
    229 076214  100557  102064  100361  EMT343: .WORD   EMS52,EMS117,EMS45,EMS57,0
    230 076226  100557  102064  100306  EMT344: .WORD   EMS52,EMS117,EMS44,EMS57,0
    231 076240  100557  102064  104614  EMT345: .WORD   EMS52,EMS117,EMS221,0
    232 076250  101612  077454  102031  EMT346: .WORD   EMS106,EMS20,EMS115,EMS223,EMS72,0
    233 076264  100557  104271  104664  EMT347: .WORD   EMS52,EMS205,EMS222,EMS206,0
    234 076276  101335  103232  101476  EMT350: .WORD   EMS75,EMS150,EMS102,EMS115,EMS56,EMS73,0
    235 076314  103644  101301  101476  EMT351: .WORD   EMS167,EMS73,EMS102,0
    236 076324  104470  000000          EMT352: .WORD   EMS215,0
    237 076330  104541  104230  104511  EMT353: .WORD   EMS217,EMS203,EMS216,0
    238 076340  104614  077454  000000  EMT354: .WORD   EMS221,EMS20,0
    239
```

```
      1 076346  105044  105650  105725  EHT1:   .WORD   EH1,STSH1,STSH2,STSH4,0
      2 076360  105650  105725  106052  EHT2:   .WORD   STSH1,STSH2,STSH4,0
      3
      4 076370  105063  000000          EHT110: .WORD   EH110,0
      5 076374  105072  000000          EHT111: .WORD   EH111,0
      6
      7 076400  105111  000000          EHT114: .WORD   EH114,0
      8 076404  105140  105650  105725  EHT223: .WORD   EH223,STSH1,STSH2,STSH4,0
      9 076416  105166  105650  105725  EHT256: .WORD   EH256,STSH1,STSH2,STSH4,0
     10
     11 076430  105242  105650  105725  EHT336: .WORD   EH336,STSH1,STSH2,STSH4,0
     12 076442  105301  105650  105725  EHT337: .WORD   EH337,STSH1,STSH2,STSH4,0
     13 076454  105436  105650  105725  EHT344: .WORD   EH344,STSH1,STSH2,STSH4,0
     14
     15 076466  105574  000000          EHT353: .WORD   EH353,0
     16
```

```
     1 076472  106110  106204  106222  EDT1:   .WORD   ED1,STSD1,STSD2,STSD4
     2 076502  106204  106222  106254  EDT2:   .WORD   STSD1,STSD2,STSD4
     3
     4 076510  106116                  EDT110: .WORD   ED110
     5 076512  106122                  EDT111: .WORD   ED111
     6
     7 076514  106130                  EDT114: .WORD   ED114
     8 076516  106140  106204  106222  EDT223: .WORD   ED223,STSD1,STSD2,STSD4
     9
    10 076526  106150  106204  106222  EDT336: .WORD   ED336,STSD1,STSD2,STSD4
    11 076536  106162  106204  106222  EDT337: .WORD   ED337,STSD1,STSD2,STSD4
    12 076546  106162  106204  106222  EDT344: .WORD   ED337,STSD1,STSD2,STSD4,0
    13
    14 076560  106174                  EDT353: .WORD   ED353
    15
```

```
      1 076562  106267  106305  106305  EFT1:   .WORD   EF111,STSF,STSF,STSF
      2 076572  106305  106305  106305  EFT2:   .WORD   STSF,STSF,STSF
      3
      4 076600  106266                  EFT110: .WORD   EF110
      5 076602  106267                  EFT111: .WORD   EF111
      6
      7 076604  106271                  EFT114: .WORD   EF114
      8 076606  106271  106305  106305  EFT223: .WORD   EF114,STSF,STSF,STSF
      9
     10 076616  106274  106305  106305  EFT336: .WORD   EF336,STSF,STSF,STSF
     11 076626  106274  106305  106305  EFT337: .WORD   EF336,STSF,STSF,STSF
     12 076636  106274  106305  106305  EFT344: .WORD   EF336,STSF,STSF,STSF
     13
     14 076646  106271                  EFT353: .WORD   EF114
     15
```

```
                                .SBTTL  ERROR MESSAGE STRINGS

076650  127  122  117  EMS1:   .ASCIZ  @WRONG UNIT SELECTED (RMCS2, BITS 0-2) @
076717  104  105  126  EMS2:   .ASCIZ  @DEVICE WENT @
076734  125  116  101  EMS3:   .ASCIZ  @UNAVAILABLE 'DVA' (RMCS1, BIT 11) @
076777  116  117  116  EMS4:   .ASCIZ  @NONEXISTENT 'NED' (RMCS2, BIT 12) @
077042  103  117  115  EMS5:   .ASCIZ  @COMMAND NOT COMPLETED, @
077072  103  117  116  EMS6:   .ASCIZ  @CONTROLLER NOT READY (RMCS1, BIT 7) @
077137  104  122  111  EMS7:   .ASCIZ  @DRIVE NOT READY 'DRY' (RMDS, BIT 7) @
077204  107  117  040  EMS10:  .ASCIZ  @GO NOT RESET 'GO' (RMCS1, BIT 0) @
077246  111  116  126  EMS11:  .ASCIZ  @INVALID @
077257  106  125  116  EMS12:  .ASCIZ  @FUNCTION CODE (RMCS1, BITS 1-5) @
077320  115  101  123  EMS13:  .ASCIZ  @MASSBUS @
077331  103  117  116  EMS14:  .ASCIZ  @CONTROL @
077342  102  125  123  EMS15:  .ASCIZ  @BUS PARITY ERROR @
077364  042  115  103  EMS16:  .ASCIZ  @'MCPE' (RMCS1, BIT 13) @
077414  124  122  101  EMS17:  .ASCIZ  @TRANSFER ERROR (RMCS1, BIT 14) @
077454  123  110  117  EMS20:  .ASCIZ  @SHOULD NOT BE SET @
077477  127  117  122  EMS21:  .ASCIZ  @WORD COUNT (RMWC) @
077522  102  125  123  EMS22:  .ASCIZ  @BUS (RMBA) @
077536  042  114  102  EMS23:  .ASCIZ  @'LBT' (RMDS, BIT 10) @
077564  042  101  117  EMS24:  .ASCIZ  @'AOE' (RMER1, BIT 09) @
077613  104  111  123  EMS25:  .ASCIZ  @DISK (RMDA) @
077630  103  131  114  EMS26:  .ASCIZ  @CYLINDER (RMDC) @
077651  101  104  104  EMS27:  .ASCIZ  @ADDRESS @
077662  123  124  101  EMS30:  .ASCIZ  @STATUS @
077672  042  127  114  EMS31:  .ASCIZ  @'WLE' (RMER1, BIT 11) @
077721  042  125  120  EMS32:  .ASCIZ  @'UPE' (RMCS2, BIT 13) @
077750  042  127  103  EMS33:  .ASCIZ  @'WCF' (RMER1, BIT 5) @
077776  127  122  111  EMS34:  .ASCIZ  @WRITE CHECK ERROR-'WCE' (RMCS2, BIT 14) @
100047  042  115  104  EMS35:  .ASCIZ  @'MDPE' (RMCS2, BIT 8) @
100076  042  104  103  EMS36:  .ASCIZ  @'DCK' (RMER1, BIT 15) @
100125  042  105  103  EMS37:  .ASCIZ  @'ECH' (RMER1, BIT 6) @
100153  042  104  114  EMS40:  .ASCIZ  @'DLT' (RMCS2, BIT 15) @
100202  042  115  130  EMS41:  .ASCIZ  @'MXF' (RMCS2, BIT 9) @
100230  042  104  124  EMS42:  .ASCIZ  @'DTE' (RMER1, BIT 12) @
100257  042  110  103  EMS43:  .ASCIZ  @'HCRC' (RMER1, BIT 8) @
100306  110  105  101  EMS44:  .ASCIZ  @HEADER COMPARE ERROR 'HCE' (RMER1, BIT 7) @
100361  106  117  122  EMS45:  .ASCIZ  @FORMAT ERROR 'FER' (RMER1, BIT 4) @
100424  042  111  101  EMS46:  .ASCIZ  @'IAE' (RMER1, BIT 10) @
100453  042  117  120  EMS47:  .ASCIZ  @'OPI' (RMER1, BIT 13) @
100502  042  123  113  EMS50:  .ASCIZ  @'SKI' (RMER2, BIT 14) @
100531  042  120  111  EMS51:  .ASCIZ  @'PIP' (RMDS, BIT 13) @
100557  124  110  105  EMS52:  .ASCIZ  @THE RM @
100567  104  105  124  EMS53:  .ASCIZ  @DETECTED @
100601  101  124  040  EMS54:  .ASCIZ  @AT AN UNEXPECTED @
100623  111  116  103  EMS55:  .ASCIZ  @INCORRECT DATA DURING @
100652  111  116  126  EMS56:  .ASCIZ  @INVALID COMMAND ERROR 'IVC' (RMER2, BIT 12) @
100727  104  125  122  EMS57:  .ASCIZ  @DURING DATA TRANSFER @
100755  104  101  124  EMS60:  .ASCIZ  @DATA READ @
100770  104  117  105  EMS61:  .ASCIZ  @DOES NOT COMPARE WITH @
101017  104  101  124  EMS62:  .ASCIZ  @DATA WRITTEN @
101035  042  120  101  EMS63:  .ASCIZ  @'PAR' (RMER1, BIT 3) @
101063  111  123  040  EMS64:  .ASCIZ  @IS INCORRECT @
101101  103  117  115  EMS65:  .ASCIZ  @COMPOSITE ERROR 'ERR' (RMDS, BIT 14) @
101147  104  101  124  EMS66:  .ASCIZ  @DATA PARITY ERROR 'DPE' (RMER2, BIT 3) @
101217  104  125  122  EMS67:  .ASCIZ  @DURING SEEK COMMAND @
```

```
 58 101244  111  123  040  EMS70:  .ASCIZ  @IS RESET @
 59 101256  105  122  122  EMS71:  .ASCIZ  @ERRONEOUS @
 60 101271  111  123  040  EMS72:  .ASCIZ  @IS SET @
 61 101301  104  111  104  EMS73:  .ASCIZ  @DID NOT SET @
 62 101316  104  111  104  EMS74:  .ASCIZ  @DID NOT RESET @
 63 101335  114  117  123  EMS75:  .ASCIZ  @LOST @
 64 101343  104  125  122  EMS76:  .ASCIZ  @DURING PACK ACK COMMAND @
 65 101374  042  122  115  EMS77:  .ASCIZ  @"RMR" (RMER1, BIT 2) @
 66 101422  042  111  114  EMS100: .ASCIZ  @"ILR" (RMER1, BIT 1) @
 67 101450  042  111  114  EMS101: .ASCIZ  @"ILF" (RMER1, BIT 0) @
 68 101476  104  125  122  EMS102: .ASCIZ  @DURING SEARCH COMMAND @
 69 101525  105  122  122  EMS103: .ASCIZ  @ERROR @
 70 101534  042  114  102  EMS104: .ASCIZ  @"LBC" (RMER2, BIT 10) @
 71 101563  042  114  123  EMS105: .ASCIZ  @"LSC" (RMER2, BIT 11) @
 72 101612  110  105  101  EMS106: .ASCIZ  @HEADER ERRORS @
 73 101631  102  125  123  EMS110: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
 74 101660  101  104  104  EMS111: .ASCIZ  @ADDRESS @
 75 101671  127  110  105  EMS112: .ASCII  @WHEN READING/WRITING RH REGISTERS @
 76 101733  101  124  040           .ASCIZ  @AT THE FOLLOWING @
 77 101755  124  110  105  EMS113: .ASCIZ  @THE SELECTED DEVICE IS @
 78 102005  116  117  116  EMS114: .ASCIZ  @NONEXISTENT DEVICE @
 79 102031  015  012  000  EMS115: .ASCIZ  <CR><LF>
 80 102034  124  110  105  EMS116: .ASCIZ  @THE MASSBUS CONTROLLER @
 81 102064  106  101  111  EMS117: .ASCIZ  @FAILED TO DETECT @
 82 102106  116  117  124  EMS120: .ASCIZ  @NOT AN RM05/3/2 @
 83 102127  103  117  116  EMS121: .ASCIZ  @CONTROL STATUS REGISTER 1, RMCS1, @
 84 102172  102  125  123  EMS122: .ASCIZ  @BUS ADDRESS REGISTER, RMBA, @
 85 102227  103  117  116  EMS123: .ASCIZ  @CONTROL STATUS REGISTER 2, RMCS2, @
 86 102272  105  122  122  EMS124: .ASCIZ  @ERROR REGISTER 1, RMER1, @
 87 102324  101  124  124  EMS125: .ASCIZ  @ATTENTION SUMMARY REGISTER, RMAS, @
 88 102367  115  101  111  EMS126: .ASCIZ  @MAINTENANCE REGISTER #1, RMMR #1, @
 89 102432  105  103  103  EMS127: .ASCIZ  @ECC POSITION REGISTER, RMEC1, @
 90 102471  105  103  103  EMS130: .ASCIZ  @ECC PATTERN REGISTER, RMEC2, @
 91 102527  115  101  111  EMS131: .ASCIZ  @MAINTENANCE REGISTER 2, RMMR2, @
 92 102567  116  117  124  EMS132: .ASCIZ  @NOT INITIALIZED BY @
 93 102613  125  116  111  EMS133: .ASCIZ  @UNIBUS INITIALIZE @
 94 102636  103  117  116  EMS134: .ASCIZ  @CONTROLLER CLEAR, I.E.  BIT 5 OF @
 95 102700  122  110  057  EMS135: .ASCIZ  @RH/RM ERROR CLEAR (RMCS1, BIT 14) @
 96 102743  104  122  111  EMS136: .ASCIZ  @DRIVE CLEAR COMMAND @
 97 102770  104  122  111  EMS137: .ASCIZ  @DRIVE STATUS REGISTER, RMDS @
 98 103025  115  105  104  EMS140: .ASCIZ  @MEDIUM OFF LINE @
 99 103046  042  115  117  EMS141: .ASCIZ  @"MOL" (RMDS, BIT 12) @
100 103074  104  122  111  EMS142: .ASCIZ  @DRIVE FAULT @
101 103111  042  104  126  EMS143: .ASCIZ  @"DVC" (RMER2, BIT 7) @
102 103137  125  116  123  EMS144: .ASCIZ  @UNSAFE @
103 103147  042  125  116  EMS145: .ASCIZ  @"UNS" (RMER1, BIT 14) @
104 103176  123  110  117  EMS146: .ASCIZ  @SHOULD BE SET @
105 103215  117  106  106  EMS147: .ASCIZ  @OFFSET MODE @
106 103232  126  117  114  EMS150: .ASCIZ  @VOLUME VALID @
107 103250  042  117  115  EMS151: .ASCIZ  @"OM" (RMDS, BIT 0) @
108 103274  042  126  126  EMS152: .ASCIZ  @"VV" (RMDS, BIT 6) @
109 103320  120  101  103  EMS153: .ASCIZ  @PACK ACK COMMAND @
110 103342  116  117  124  EMS154: .ASCIZ  @NOT SET BY @
111 103356  117  106  106  EMS155: .ASCIZ  @OFFSET COMMAND @
112 103376  116  117  124  EMS156: .ASCIZ  @NOT RESET BY @
113 103414  122  124  103  EMS157: .ASCIZ  @RTC COMMAND @
114 103431  122  111  120  EMS160: .ASCIZ  @RIP COMMAND @
```

```
115 103446      117     106     106  EMS161: .ASCIZ  @OFFSET REGISTER (RMOF) @
116 103476                           EMS162:         ;<UNUSED>
117 103476      104     125     122  EMS163: .ASCIZ  @DURING RECALIBRATE @
118 103522      111     123     040  EMS164: .ASCII  @IS INTERMITTENT OR DRIVE DIDNT DROP ON @
119 103571      103     131     114          .ASCIZ  @CYLINDER @
120 103603      125     116     105  EMS165: .ASCIZ  @UNEXPECTED @
121 103617      122     105     103  EMS166: .ASCIZ  @RECALIBRATE COMMAND @
122 103644      042     101     124  EMS167: .ASCIZ  @'ATA'' (RMDS, BIT15) @
123 103671      127     110     105  EMS170: .ASCIZ  @WHEN READING REGISTER @
124 103720      105     122     122  EMS171: .ASCIZ  @ERROR REGISTER #2, RMER2, @
125 103753      116     117     116  EMS172: .ASCIZ  @NONRECOVERABLE @
126 103773      122     105     103  EMS173: .ASCIZ  @RECOVERABLE @
127 104010      104     101     124  EMS174: .ASCIZ  @DATA @
128 104016      104     125     122  EMS175: .ASCIZ  @DURING WRITE COMMAND @
129 104044      042     117     120  EMS176: .ASCIZ  @'DPE'' (RMER2, BIT 13) @
130 104073      111     116     040  EMS177: .ASCIZ  @IN WRITE PROTECT @
131 104115      103     101     116  EMS200: .ASCIZ  @CAN NOT SET @
132 104132      104     111     101  EMS201: .ASCIZ  @DIAGNOSTIC MODE 'DMD'' (RMMR1, BIT 0) @
133 104200      104     125     122  EMS202: .ASCIZ  @DURING DIAGNOSTIC MODE @
134 104230      111     116     103  EMS203: .ASCIZ  @INCORRECT @
135 104243      042     127     122  EMS204: .ASCIZ  @'WRL'' (RMDS, BIT 11) @
136 104271      105     130     105  EMS205: .ASCIZ  @EXECUTED @
137 104303      127     111     124  EMS206: .ASCIZ  @WITH COMP ERROR SET @
138 104330      042     107     117  EMS207: .ASCIZ  @'GO'' (RMCS1, BIT 0) @
139 104355      127     122     111  EMS210: .ASCIZ  @WRITING @
140 104366      127     101     123  EMS211: .ASCIZ  @WAS RESET BY @
141 104404      120     122     117  EMS212: .ASCIZ  @PROGRAM INTERRUPT @
142 104427      127     101     123  EMS213: .ASCIZ  @WAS NOT GENERATED @
143 104452      123     105     105  EMS214: .ASCIZ  @SEEK COMMAND @
144 104470      120     122     117  EMS215: .ASCIZ  @PROGRAM TIMEOUT @
145 104511      104     125     122  EMS216: .ASCIZ  @DURING LOOK AHEAD TEST @
146 104541      114     117     117  EMS217: .ASCIZ  @LOOK AHEAD REGISTER,RMLA, @
147 104574      123     105     101  EMS220: .ASCIZ  @SEARCH COMMAND @
148 104614      102     101     104  EMS221: .ASCIZ  @BAD SECTOR ERROR 'BSE'' (RMER2, BIT 15) @
149 104664      101     040     104  EMS222: .ASCIZ  @A DATA TRANSFER COMMAND @
150 104715      110     105     101  EMS223: .ASCIZ  @HEADER COMPARE INHIBIT 'HCI'' (RMOF, BIT 10) @
151 104772      116     117     116  EMS224: .ASCIZ  @NONEXISTENT MEMORY 'NEM'' (RMCS2, BIT 11) @
152
```

ERROR MESSAGE STRINGS

SEQ 0291

```
      1 105044      105      130      120  EH1:    .ASCIZ  @EXPCTD  RECEVD@
      2 105063      102      125      123  EH110:  .ASCIZ  @BUSADR@
      3 105072      040      122      115  EH111:  .ASCIZ  @ RMCS2   RMCS1@
      4
      5 105111      122      105      103  EH114:  .ASCIZ  @RECEVD  SNGPRT  DULPRT@
      6 105140      105      130      120  EH223:  .ASCIZ  @EXPCTD  RECEVD   DATA@
      7 105166      105      130      120  EH256:  .ASCII  @EXPCTD  RECEVD  RGSTR@<CRLF>
      8 105214      123      124      101          .ASCIZ  @STATUS  STATUS  INDEX@
      9
     10 105242      107      104      101  EH336:  .ASCIZ  @GDADRS  GDDATA  BDADRS  BDDATA@
     11 105301      122      115      103  EH337:  .ASCII  @RMCS2   STATUS  FAILING   DATA@<CRLF>
     12 105340      137      137      137          .ASCII  @                            @<CRLF>
     13 105377      105      130      120          .ASCIZ  @EXPCTD--RECEVD  --BIT---ADRESS@
     14
     15 105436      122      115      105  EH344:  .ASCII  @RMER1   STATUS  HEADER  FAILING@<CRLF>
     16 105476      137      137      137          .ASCII  @                WORD     BIT@<CRLF>
     17 105534      105      130      120          .ASCIZ  @EXPCTD--RECEVD  NUMBER  POSITON@
     18 105574      105      130      120  EH353:  .ASCII  @EXPCTD  RECEVD@<CRLF>
     19 105613      074      103      122          .ASCIZ  @<CRLF> RMLA    RMLA    RMOF @
     20
     21 105650      040      122      115  STSH1:  .ASCII  @ RMCS1  RMCS2   RMDS    RMER1   RMER2@
     22 105715      040      040      040          .ASCIZ  @   RMAS@
     23 105725      040      122      115  STSH2:  .ASCII  @ RMWC    RMBA    RMDA    RMOF    RMDC@
     24 105772      040      040      040          .ASCIZ  @    RMEC1   RMEC2@
     25 106014      040      122      115  STSH3:  .ASCIZ  @ RMDA    RMDC    RMOF    RMLA@
     26 106052      040      122      115  STSH4:  .ASCIZ  @ RMMR1   RMMR2   RMDT    RMSN@
     27                                   .EVEN
     28
```

```
      1 106110  001140  001142  000000  ED1:    .WORD   $GDDAT,$BDDAT,0
      2 106116  001276  000000          ED110:  .WORD   $BASE,0
      3 106122  001174  001176  000000  ED111:  .WORD   $TMP0,$TMP1,0
      4
      5 106130  001362  001176  001200  ED114:  .WORD   RMDTI,$TMP1,$TMP2,0
      6 106140  001140  001142  001174  ED223:  .WORD   $GDDAT,$BDDAT,$TMP0,0
      7
      8 106150  001134  001140  001136  ED336:  .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
      9
     10 106162  001140  001142  001174  ED337:  .WORD   $GDDAT,$BDDAT,$TMP0,$TMP1,0
     11 106174  001140  001142  001442  ED353:  .WORD   $GDDAT,$BDDAT,RMOF0,0
     12
     13 106204  001334  001344  001346  STSD1:  .WORD   RMCS1I,RMCS2I,RMDSI,RMER1I,RMER2I,RMASI,0
     14 106222  001336  001340  001342  STSD2:  .WORD   RMWCI,RMBAI,RMDAI,RMOFI,RMDCI,RMEC1I
     15 106236  001402  000000          .WORD   RMEC2I,0
     16 106242  001342  001370  001366  STSD3:  .WORD   RMDAI,RMDCI,RMOFI,RMLAI,0
     17 106254  001360  001374  001362  STSD4:  .WORD   RMMR1I,RMMR2I,RMDTI,RMSNI,0
     18
```

```
                                                                J 7
      1 106266  000                       EF110:  .BYTE   0
      2 106267  000   000                 EF111:  .BYTE   0,0
      3 106271  000   000   000           EF114:  .BYTE   0,0,0
      4 106274  000   000   000           EF336:  .BYTE   0,0,0,0
      5 106300  000   000   000           EF337:  .BYTE   0,0,0,0,0
      6
      7 106305  000   000   000           STSF:   .BYTE   0,0,0,0,0,0,0
      8                                   .EVEN
      9
```

```
                                ;STORAGE FOR GENERAL DATA TRANSFERRS
106314                          BUFFER:
106314                          BUFONE: .BLKW   258.
107320                          BUFTWO: .BLKW   258.

                                ;STORAGE FOR MANUFACTURES 16 BIT MODE BAD SECTOR FILE
110324  000000  000000        MFGFIL: .WORD   0,0              ;2 HEADER WORDS
                                        .BLKW   256.            ;256. WORDS OF DATA
111330  177777                        .WORD   -1              ;TERMINATOR IF FILE IS FULL

                                ;STORAGE FOR USERS 16 BIT MODE BAD SECTOR FILE
111332  000000  000000        USRFIL: .WORD   0,0              ;2 HEADER WORDS
                                        .BLKW   256.            ;256. WORDS OF DATA
112336  177777                        .WORD   -1              ;TERMINATOR IF FILE IS FULL

        106314                  .=BUFFER

106314                          HELP:
106314    200                   .ASCII  <CRLF>
106315    200                   .ASCII  <CRLF>
106316    114     111     123   .ASCII  @LIST OF TESTS@<CRLF>
106334    055     055     055   .ASCII  @-------------@<CRLF>
106352    124     061     011   .ASCII  @T1     CONTROLLER ACCESS TEST@<CRLF>
106404    124     062     011   .ASCII  @T2     DEVICE AVAILABLE TEST@<CRLF>
106435    124     063     011   .ASCII  @T3     DRIVE TYPE TEST@<CRLF>
106460    124     064     011   .ASCII  @T4     WRITE, READ ZEROS@<CRLF>
106505    124     065     011   .ASCII  @T5     WRITE, WRITE CHECK ZEROS@<CRLF>
106541    124     066     011   .ASCII  @T6     WRITE, WRITE CHECK ZEROS W/ WCE ERROR@<CRLF>
106612    124     067     011   .ASCII  @T7     WRITE, READ ONES@<CRLF>
106636    124     061     060   .ASCII  @T10    WRITE, WRITE CHECK ONES@<CRLF>
106672    124     061     061   .ASCII  @T11    WRITE, WRITE CHECK ONES W/ WCE ERROR@<CRLF>
106743    124     061     062   .ASCII  @T12    WRITE, WRITE CHECK MULTIPLE SECTORS@<CRLF>
107013    124     061     063   .ASCII  @T13    WRITE, READ W/ IMPLIED SEEK@<CRLF>
107053    124     061     064   .ASCII  @T14    WRITE, WRITE CHECK W/ HEAD SWITCHING@<CRLF>
107124    124     061     065   .ASCII  @T15    WRITE, WRITE CHECK W/ MID-TRANSFER SEEK@<CRLF>
107200    124     061     066   .ASCII  @T16    WRITE, READ W/ HCE ERROR@<CRLF>
107235    124     061     067   .ASCII  @T17    WRITE, READ W/ HCI@<CRLF>
107264    124     062     060   .ASCII  @T20    WRITE, READ W/ IVC ERROR@<CRLF>
107321    124     062     061   .ASCII  @T21    WRITE, READ W/ ABORT@<CRLF>
107352    124     062     062   .ASCII  @T22    WRITE, READ EACH CURRENT LEVEL@<CRLF>
107415    124     062     063   .ASCII  @T23    WRITE, WRITE CHECK W/ CURRENT LEVEL SWITCHING@<CRLF>
107477    124     062     064   .ASCII  @T24    WRITE, READ EARLY PEAK SHIFT@<CRLF>
107540    124     062     065   .ASCII  @T25    WRITE, READ MIXED PEAK SHIFT@<CRLF>
107601    124     062     066   .ASCII  @T26    WRITE, READ EACH TRACK@<CRLF>
107634    124     062     067   .ASCII  @T27    READ, WRITE CHECK MULTIPLE SECTORS IN OFFSET MODE@<CRLF>
107722    200                   .ASCII  <CRLF>
107723    117     120     105   .ASCII  @OPERATIONAL SWITCH SETTINGS@<CRLF>
107757    055     055     055   .ASCII  @-------------------------@<CRLF>
110013    123     127     111   .ASCII  @SWITCH                   USE@<CRLF>
110030    055     055     055   .ASCII  @------          ----------------@<CRLF>
110065    040     040     061   .ASCII  @  15            HALT ON ERROR@<CRLF>
110111    040     040     061   .ASCII  @  14            LOOP ON TEST@<CRLF>
110134    040     040     061   .ASCII  @  13            INHIBIT ERROR TYPEOUTS@<CRLF>
110171    040     040     061   .ASCII  @  12            @<CRLF>
110200    040     040     061   .ASCII  @  11            INHIBIT ITERATIONS@<CRLF>
110231    040     040     061   .ASCII  @  10            BELL ON ERROR@<CRLF>
110255    040     040     040   .ASCII  @   9            LOOP ON ERROR@<CRLF>
```

```
     58 110301      040      040      040  .ASCII  @   8          LOOP ON TEST IN SWR<7:0>@<CRLF>
     59 110340      040      040      040  .ASCII  @   7          TN128@<CRLF>
     60 110354      040      040      040  .ASCII  @   6          TN64@<CRLF>
     61 110367      040      040      040  .ASCII  @   5          TN32@<CRLF>
     62 110402      040      040      040  .ASCII  @   4          TN16@<CRLF>
     63 110415      040      040      040  .ASCII  @   3          TN8@<CRLF>
     64 110427      040      040      040  .ASCII  @   2          TN4@<CRLF>
     65 110441      040      040      040  .ASCIZ  @   1          TN2@<CRLF>
     66
     67          000200                   .END    200
```

SYMBOL TABLE

```
ABASE = 176700        ATNMSK= 000377        CHRCNT  040433        EDT1    076472        EMS100  101422
ACDW1 = 000000        ATNTBL  071616        CKSWR = 104410        EDT110  076510        EMS101  101450
ACDW2 = 000000        AUNIT = 000000        CLKADR  001516        EDT111  076512        EMS102  101476
ACKSTS  054602        AUSWR = 000000        CLKVCT  001520        EDT114  076514        EMS103  101525
ACPUOP= 000000        AVECT1= 120254        CLR   = 000040        EDT2    076502        EMS104  101534
ADDW0 = 000000        AVECT2= 000000        CLRSTS  053722        EDT223  076516        EMS105  101563
ADDW1 = 000000        A16   = 000400        CMNSTA  007612        EDT336  076526        EMS106  101612
ADDW10= 000000        A17   = 001000        CMPBUF  043602        EDT337  076536        EMS11   077246
ADDW11= 000000        BACK  = 000001        CMPERR  051720        EDT344  076546        EMS110  101631
ADDW12= 000000        BADSCT  041416        CNSL00  070755        EDT353  076560        EMS111  101660
ADDW13= 000000        BADTMO  005340        CNSL01  071010        ED1     106110        EMS112  101671
ADDW14= 000000        BAI   = 000010        CNSL02  071026        ED110   106116        EMS113  101755
ADDW15= 000000        BB00  = 000001        CNSL03  071070        ED111   106122        EMS114  102005
ADDW2 = 000000        BB01  = 000002        CNSL04  071110        ED114   106130        EMS115  102031
ADDW3 = 000000        BB02  = 000004        CNSL05  071144        ED223   106140        EMS116  102034
ADDW4 = 000000        BB03  = 000010        CNSL06  071156        ED336   106150        EMS117  102064
ADDW5 = 000000        BB04  = 000020        CNSL07  071207        ED337   106162        EMS12   077257
ADDW6 = 000000        BB05  = 000040        CNSL08  071352        ED353   106174        EMS120  102106
ADDW7 = 000000        BB06  = 000100        CNSL09  071353        EECC  = 000020        EMS121  102127
ADDW8 = 000000        BB07  = 000200        CNTCLR  053604        EFT1    076562        EMS122  102172
ADDW9 = 000000        BB08  = 000400        COMMA   070342        EFT110  076600        EMS123  102227
ADEVCT= 000000        BB09  = 001000        CONT  = 000100        EFT111  076602        EMS124  102272
ADEVM = 000000        BIT0  = 000001        CR    = 000015        EFT114  076604        EMS125  102324
ADR   = 000001        BIT00 = 000001        CRLF  = 000200        EFT2    076572        EMS126  102367
AENV  = 000000        BIT01 = 000002        CTLFG   001326        EFT223  076606        EMS127  102432
AENVM = 000000        BIT02 = 000004        CYLMSK= 001777        EFT336  076616        EMS13   077320
AFATAL= 000000        BIT03 = 000010        DBCK  = 100000        EFT337  076626        EMS130  102471
ALL     070331        BIT04 = 000020        DBEN  = 040000        EFT344  076636        EMS131  102527
AMADR1= 000000        BIT05 = 000040        DBL   = 002000        EFT353  076646        EMS132  102567
AMADR2= 000000        BIT06 = 000100        DCK   = 100000        EF110   106266        EMS133  102613
AMADR3= 000000        BIT07 = 000200        DDISP = 177570        EF111   106267        EMS134  102636
AMADR4= 000000        BIT08 = 000400        DEBL  = 020000        EF114   106271        EMS135  102700
AMAMS1= 000000        BIT09 = 001000        DEVSEL  052132        EF336   106274        EMS136  102743
AMAMS2= 000000        BIT1  = 000002        DISPLA  001156        EF337   106300        EMS137  102770
AMAMS3= 000000        BIT10 = 002000        DISPRE  000174        EHT1    076346        EMS14   077331
AMAMS4= 000000        BIT11 = 004000        DLT   = 100000        EHT110  076370        EMS140  103025
AMSGAD= 000000        BIT12 = 010000        DMD   = 000001        EHT111  076374        EMS141  103046
AMSGLG= 000000        BIT13 = 020000        DPE   = 000010        EHT114  076400        EMS142  103074
AMSGTY= 000000        BIT14 = 040000        DPEHI = 040000        EHT2    076360        EMS143  103111
AMTYP1= 000000        BIT15 = 100000        DPELO = 020000        EHT223  076404        EMS144  103137
AMTYP2= 000000        BIT2  = 000004        DPR   = 000400        EHT256  076416        EMS145  103147
AMTYP3= 000000        BIT3  = 000010        DRQ   = 004000        EHT336  076430        EMS146  103176
AMTYP4= 000000        BIT4  = 000020        DRVCLR= 000010        EHT337  076442        EMS147  103215
AOE   = 001000        BIT5  = 000040        DRVSTS  057140        EHT344  076454        EMS15   077342
APASS = 000000        BIT6  = 000100        DRY   = 000200        EHT353  076466        EMS150  103232
APE   = 100000        BIT7  = 000200        DSWR  = 177570        EH1     105044        EMS151  103250
APRIOR= 000000        BIT8  = 000400        DTASTS  057742        EH110   105063        EMS152  103274
APTCSU= 000040        BIT9  = 001000        DTE   = 010000        EH111   105072        EMS153  103320
APTENV= 000001        BOTADR  040430        DTO   = 010000        EH114   105111        EMS154  103342
APTSIZ= 000200        BOTFLG  040432        DULPRT= 024024        EH223   105140        EMS155  103356
APTSPO= 000100        BPTVEC= 000014        DVA   = 004000        EH256   105166        EMS156  103376
ARGS  = 000004        BSE   = 100000        DVC   = 000200        EH336   105242        EMS157  103414
ASNDA   001514        BUFFER  106314        EARLY   072142        EH337   105301        EMS16   077364
ASNDC   001512        BUFONE  106314        EBL   = 020000        EH344   105436        EMS160  103431
ASWREG= 000000        BUFTWO  107320        ECH   = 000100        EH353   105574        EMS161  103446
ATA   = 100000        CC    = 004000        ECI   = 004000        EMS1    076650        EMS162  103476
ATESTN= 000000        CH    = 002000        ECRC  = 001000        EMS10   077204        EMS163  103476
```

EMS164	103522	EMS44	100306	EMT125	073626	EMT207	074530	EMT271	075276
EMS165	103603	EMS45	100361	EMT126	073636	EMT21	072420	EMT272	075314
EMS166	103617	EMS46	100424	EMT127	073650	EMT210	074542	EMT273	075332
EMS167	103644	EMS47	100453	EMT13	072342	EMT211	074550	EMT274	075350
EMS17	077414	EMS5	077042	EMT130	073662	EMT212	074564	EMT275	075366
EMS170	103671	EMS50	100502	EMT131	073674	EMT213	074600	EMT276	075400
EMS171	103720	EMS51	100531	EMT132	073706	EMT214	074610	EMT277	075414
EMS172	103753	EMS52	100557	EMT133	073720	EMT215	074630	EMT3	072266
EMS173	103773	EMS53	100567	EMT134	073732	EMT216	074642	EMT30	072510
EMS174	104010	EMS54	100601	EMT135	073744	EMT217	074652	EMT300	075432
EMS175	104016	EMS55	100623	EMT136	073756	EMT22	072430	EMT301	075452
EMS176	104044	EMS56	100652	EMT137	073770	EMT220	074662	EMT302	075474
EMS177	104073	EMS57	100727	EMT14	072354	EMT221	074672	EMT303	075506
EMS2	076717	EMS6	077072	EMT140	074000	EMT222	074702	EMT304	075520
EMS20	077454	EMS60	100755	EMT141	074010	EMT223	074712	EMT305	075532
EMS200	104115	EMS61	100770	EMT142	074020	EMT224	074722	EMT306	075544
EMS201	104132	EMS62	101017	EMT143	074030	EMT225	074724	EMT307	075554
EMS202	104200	EMS63	101035	EMT144	074040	EMT226	074726	EMT31	072520
EMS203	104230	EMS64	101063	EMT145	074050	EMT227	074730	EMT310	075574
EMS204	104243	EMS65	101101	EMT146	074060	EMT23	072440	EMT311	075606
EMS205	104271	EMS66	101147	EMT147	074070	EMT230	074732	EMT312	075620
EMS206	104303	EMS67	101217	EMT15	072362	EMT231	074734	EMT313	075632
EMS207	104330	EMS7	077137	EMT150	074100	EMT232	074736	EMT314	075652
EMS21	077477	EMS70	101244	EMT151	074110	EMT233	074740	EMT315	075664
EMS210	104355	EMS71	101256	EMT152	074122	EMT234	074742	EMT316	075676
EMS211	104366	EMS72	101271	EMT153	074134	EMT235	074744	EMT317	075710
EMS212	104404	EMS73	101301	EMT154	074152	EMT236	074746	EMT32	072530
EMS213	104427	EMS74	101316	EMT155	074170	EMT237	074750	EMT320	075720
EMS214	104452	EMS75	101335	EMT156	074202	EMT24	072450	EMT321	075730
EMS215	104470	EMS76	101343	EMT157	074214	EMT240	074752	EMT322	075740
EMS216	104511	EMS77	101374	EMT16	072370	EMT241	074754	EMT323	075746
EMS217	104541	EMTVEC=	000030	EMT160	074226	EMT242	074756	EMT324	075756
EMS22	077522	EMT1	072254	EMT161	074236	EMT243	074760	EMT325	075770
EMS220	104574	EMT10	072324	EMT162	074250	EMT244	074762	EMT326	076002
EMS221	104614	EMT100	073322	EMT163	074262	EMT245	074764	EMT327	076020
EMS222	104664	EMT101	073340	EMT164	074272	EMT246	074766	EMT33	072540
EMS223	104715	EMT102	073356	EMT165	074300	EMT247	074776	EMT330	076032
EMS224	104772	EMT103	073366	EMT166	074306	EMT25	072460	EMT331	076042
EMS23	077536	EMT104	073400	EMT167	074314	EMT250	075010	EMT332	076054
EMS24	077564	EMT105	073416	EMT17	072400	EMT251	075024	EMT333	076066
EMS25	077613	EMT106	073426	EMT170	074322	EMT252	075032	EMT334	076104
EMS26	077630	EMT107	073436	EMT171	074332	EMT253	075040	EMT335	076122
EMS27	077651	EMT11	072330	EMT172	074342	EMT254	075056	EMT336	076142
EMS3	076734	EMT110	073456	EMT173	074352	EMT255	075066	EMT337	076152
EMS30	077662	EMT111	073470	EMT174	074362	EMT256	075076	EMT34	072550
EMS31	077672	EMT112	073476	EMT175	074374	EMT257	075106	EMT340	076162
EMS32	077721	EMT113	073504	EMT176	074402	EMT26	072470	EMT341	076174
EMS33	077750	EMT114	073520	EMT177	074410	EMT260	075120	EMT342	076202
EMS34	077776	EMT115	073526	EMT2	072260	EMT261	075132	EMT343	076214
EMS35	100047	EMT116	073536	EMT20	072410	EMT262	075152	EMT344	076226
EMS36	100076	EMT117	073546	EMT200	074422	EMT263	075162	EMT345	076240
EMS37	100125	EMT12	072334	EMT201	074434	EMT264	075172	EMT346	076250
EMS4	076777	EMT120	073556	EMT202	074446	EMT265	075212	EMT347	076264
EMS40	100153	EMT121	073566	EMT203	074460	EMT266	075232	EMT35	072560
EMS41	100202	EMT122	073576	EMT204	074472	EMT267	075244	EMT350	076276
EMS42	100230	EMT123	073606	EMT205	074510	EMT27	072500	EMT351	076314
EMS43	100257	EMT124	073616	EMT206	074520	EMT270	075262	EMT352	076324

```
EMT353  076330    FNCMSK= 000077    IR     = 000100    PHA    = 000200    RMDAI   001342
EMT354  076340    F0     = 000002   IVC    = 010000    PIP    = 020000    RMDAO   001416
EMT36   072570    F1     = 000004   LBC    = 002000    PIRQ   = 177772    RMDB   = 000022
EMT37   072600    F2     = 000010   LBT    = 002000    PIRQVE= 000240    RMDBI   001356
EMT4    072274    F3     = 000020   LF     = 000012    PLFS   = 002000    RMDBO   001432
EMT40   072610    F4     = 000040   LODEV    071416    PRIERR  045226    RMDC   = 000034
EMT41   072616    GENBUF   043344   LS     = 000004    PR0    = 000000    RMDCI   001370
EMT42   072626    GET      044230   LSC    = 004000    PR1    = 000040    RMDCO   001444
EMT43   072640    GETBUF   001334   LST    = 000002    PR2    = 000100    RMDS   = 000012
EMT44   072650    GETINX   001522   LSTRK    001332    PR3    = 000140    RMDSI   001346
EMT45   072660    GETSTS   044144   MCLK   = 004000    PR4    = 000200    RMDSO   001422
EMT46   072670    GO     = 000001   MCPE   = 020000    PR5    = 000240    RMDT   = 000026
EMT47   072700    GTSWR  = 104407   MDF    = 000100    PR6    = 000300    RMDTI   001362
EMT5    072302    HCE    = 000200   MDPE   = 000400    PR7    = 000340    RMDTO   001436
EMT     072706    HCI    = 002000   MEDENB   001510    PS     = 177776    RMEC1 = 000044
EMT51   072716    HCRC   = 000400   MFGFIL   110324    PSEL   = 002000    RMEC1I  001400
EMT52   072730    HELP     106314   MI     = 000004    PSW    = 177776    RMEC1O  001454
EMT53   072746    HT     = 000011   MIXED    071626    PUT      044500    RMEC2 = 000046
EMT54   072764    IAE    = 002000   MOC    = 000400    PUTBUF   001410    RMEC2I  001402
EMT55   072774    IDXMSK= 000077    MOH    = 020000    PUTINX   001551    RMEC2O  001456
EMT56   073006    IE     = 000100   MOL    = 010000    PWRVEC= 000024    RMER1 = 000014
EMT57   073024    ILF    = 000001   MRD    = 002000    QUES     070336    RMER1I  001350
EMT6    072310    ILF02  = 000002   MS     = 000040    RCLSTS   055376    RMER1O  001424
EMT60   073034    ILF24  = 000024   MSC    = 000002    RD     = 000070    RMER2 = 000042
EMT61   073052    ILF26  = 000026   MSDRVS   071374    RDCHR  = 104411    RMER2I  001376
EMT62   073072    ILF30  = 000030   MSE    = 100000    RDLIN  = 104412    RMER2O  001452
EMT63   073106    ILF32  = 000032   MSER   = 000200    RDOCT  = 104413    RMHR   = 000036
EMT64   073110    ILF34  = 000034   MSGDRV   071410    RDY    = 000200    RMHRI   001372
EMT65   073130    ILF36  = 000036   MSHELP   070345    READY    007764    RMHRO   001446
EMT66   073150    ILF40  = 000040   MUR    = 001000    RECAL  = 000006    RMLA   = 000020
EMT67   073162    ILF42  = 000042   MWD    = 000010    RESREG= 104415    RMLAI   001354
EMT7    072316    ILF44  = 000044   MWP    = 000010    RESVEC= 000010    RMLAO   001430
EMT70   073174    ILF46  = 000046   MXF    = 001000    REX    = 010000    RMMR1 = 000024
EMT71   073204    ILF54  = 000054   NDTMSK= 115760    RG     = 040000    RMMR1I  001360
EMT72   073214    ILF56  = 000056   NED    = 010000    RGDTPT   071626    RMMR1O  001434
EMT73   073232    ILF64  = 000064   NEM    = 004000    RH     = 000072    RMMR2 = 000040
EMT74   073250    ILF66  = 000066   NOP    = 000000    RIP    = 000020    RMMR2I  001374
EMT75   073260    ILF74  = 000074   NOTAVL   071453    RLEASE= 000012    RMMR2O  001450
EMT76   073300    ILF76  = 000076   NOTPRS   071436    RMAS   = 000016    RMOF   = 000032
EMT77   073310    ILR    = 000002   NSA    = 100000    RMASI   001352    RMOFI   001366
ENRGDT  072254    ILRG50= 000050    OCC    = 100000    RMASO   001426    RMOFO   001442
EQUALS  070327    ILRG52= 000052    OFD    = 000200    RMBA   = 000004    RMR    = 000004
ERR    = 040000   ILRG54= 000054    OFFLIN   071472    RMBAE  = 000050    RMSN   = 000030
ERRNMB  040426    ILRG56= 000056    OFFSET= 000014    RMBAEI  001404    RMSNI   001364
ERROR  = 104000   ILRG60= 000060    OM     = 000001    RMBAEO  001460    RMSNO   001440
ERRTYP  037670    ILRG62= 000062    ONES     071666    RMBAI   001340    RMWC   = 000002
ERRVEC= 000004    ILRG64= 000064    ONLINE   071504    RMBAO   001414    RMWCI   001336
ERTY00  040434    ILRG66= 000066    OPE    = 020000    RMCS1  = 000000    RMWCO   001412
ERTY01  040442    ILRG70= 000070    OPI    = 020000    RMCS1I  001334    RQA    = 100000
ERTY02  040452    ILRG72= 000072    OR     = 000200    RMCS10  001410    RQB    = 040000
ERTY03  040461    ILRG74= 000074    PACACK= 000022    RMCS2  = 000010    RTC    = 000016
ERTY04  040467    ILRG76= 000076    PAKACK= 000022    RMCS2I  001344    R6     =%000006
ESRC   = 004000   IOTVEC= 000020    PAR    = 000010    RMCS20  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    RMCS30  001462    SA1    = 000001
FNCDTB  071516    IPCK3  = 000010    PGM    = 001000    RMDA   = 000006    SA16   = 000020
```

SYMBOL TABLE

```
SA2    = 000002
SA4    = 000004
SA8    = 000010
SC     = 100000
SCOPE  = 000004
SCTMSG  070230
SCTMSK= 003700
SC0    = 000100
SC1    = 000200
SC2    = 000400
SC3    = 001000
SC4    = 002000
SEARCH= 000030
SECERR  046060
SEEK   = 000004
SEKSTS  052344
SHUT    064076
SHUT2   064154
SIZCLK  044720
SKI    = 040000
SNGPRT= 020024
STACK  = 001100
STANDA  006706
START   005420
STCDRV  063342
STKLMT= 177774
STOP    064046
STSD1   106204
STSD2   106222
STSD3   106242
STSD4   106254
STSF    106305
STSH1   105650
STSH2   105725
STSH3   106014
STSH4   106052
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  045042
TKVEC  = 000060
TPVEC  = 000064
TRAPVE= 000034
TRE    = 040000
TRTVEC= 000014
TST    = 010000
TSTNMB  040424
TSTPRP  040472
TSTQUE  001464
TST1    010130
TST10   015070
TST11   016046
TST12   017244
TST13   020250
TST14   021474
TST15   022702
TST16   023724
TST17   025572
TST2    010314
TST20   027176
TST21   030216
TST22   031134
TST23   032172
TST24   033176
TST25   034204
TST26   035212
TST27   036240
TST3    010500
TST4    010702
TST5    011710
TST6    012666
TST7    014062
TYPBN  = 104406
TYPDS  = 104405
TYPE   = 104401
TYPOC  = 104402
TYPON  = 104404
TYPOS  = 104403
UBUSQS  070722
UNS    = 040000
UNTMSK= 000007
UPE    = 020000
USE    = 040000
USRFIL  111332
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   071730
$APTHD  001100
$ATYC   070006
$ATY1   067762
$ATY3   067770
$ATY4   070000
$AUTOB  001150
$BASE   001276
$BDADR  001136
$BDDAT  001142
$BELL   001212
$BIN    064332
$CDW1   001302
$CDW2   001304
$CHARC  065336
$CKSWR  066434
$CMTAG  001114
$CM3   = 000000
$CM4   = 000005
$CNTLC  067332
$CNTLG  067344
$CNTLU  067337
$CPUOP  001250
$CRLF   001217
$DBLK   064550
$DDW0   001306
$DDW1   001310
$DDW2   001312
$DDW3   001314
$DDW4   001316
$DDW5   001320
$DDW6   001322
$DDW7   001324
$DEVCT  001232
$DEVM   001300
$DOAGN  037660
$DTBL   064540
$ENDAD  037650
$ENDCT  037514
$ENULL  037664
$ENV    001242
$ENVM   001243
$EOP    037460
$EOPCT  037506
$EOSP   037422
$ERFLG  001117
$ERMAX  001131
$ERROR  065736
$ERRPC  001132
$ERRTB  001600
$ERTTL  001126
$ESCAP  001210
$ETABL  001242
$ETEND  001326
$FATAL  001224
$FFLG   070226
$FILLC  001172
$FILLS  001171
$GDADR  001134
$GDDAT  001140
$GET42  037640
$GTSWR  066524
$HD    = 000000
$HIBTS  001100
$HIOCT  067474
$ICNT   001120
$ILLUP  067744
$INTAG  001151
$ITEMB  001130
$LF     001220
$LFLG   070225
$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   070224
$MNEW   067362
$MSGAD  001236
$MSGLG  001240
$MSGTY  001222
$MSWR   067351
$MTYP1  001253
$MTYP2  001257
$MTYP3  001263
$MTYP4  001267
$MXCNT  065656
$NULL   001170
$NWTST= 000001
$OCNT   065002
$OMODE  065004
$OVER   065642
$PASS   001230
$PASTM  001106
$POWER  067752
$PWRDN  067604
$PWRMG  067740
$PWRUP  067656
$QUES   001216
$RDCHR  066776
$RDLIN  067066
$RDOCT  067374
$RDSZ  = 000010
$RESPE  064222
$RTNAD  037662
$SAVRE  064164
$SAVR6  067750
$SCOPE  065342
$SETUP= 000137
$SIUP  = 177777
$SVLAD  065606
$SVPC  = 000204
$SWR   = 167400
$SWREG  001244
$SWRMK= 000000
$SW08T  065660
$TESTN  001226
$TIMES  001206
$TKB    001162
$TKCNT  066136
$TKINT  066146
$TKQEN= 066145
$TKQIN  066140
$TKQOU  066142
$TKQSR  066144
$TKS    001160
$TKSRV  066216
$TMP0   001174
$TMP1   001176
$TMP2   001200
$TMP3   001202
$TMP4   001204
$TN    = 000030
$TPB    001166
$TPFLG  001173
$TPS    001164
$TRAP   067476
$TRAP2  067536
$TRP   = 000016
$TRPAD  067550
$TSTM   001104
$TSTNM  001116
```

```
$TTYIN  067322          $TYPEX  065340          $UNIT   001234          $VECT2  001274          $$GET4= 000000
$TYPBN  064260          $TYPOC  064604          $UNITM  001110          $XOFF = 000023          $$SW08= 000030
$TYPDS  064334          $TYPON  064620          $USWR   001246          $XON  = 000021          $OFILL  065003
$TYPE   065006          $TYPOS  064560          $VECT1  001272          $XTSTR  065360          .$X   = 001100
$TYPEC  065220

. ABS.  112340      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  55976 WORDS  ( 219 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  69 PAGES
CZRMOA.BIC,CZRMOA/C=CZRMOA.DOC,CZRMOA,SYSMAC/M
```

CROSS REFERENCE TABLE (CREF V01-05)

SEQ 0302

```
$DDW4    5-0#
$DDW5    5-0#
$DDW6    5-0#
$DDW7    5-0#
$DEVCT   5-0#
$DEVM    5-0#     9-64*    9-74     9-96*    10-115*  10-122*  10-149*  10-150   11-3     11-30
$DOAGN  13-20    13-20    13-20#
$DTBL   38-3     38-3#
$ENDAD   4-786    9-25    13-20#   38-7
$ENDCT   9-19    13-20#
$ENULL  13-20#
$ENV     5-0#     9-25    11-27    38-5     38-7     38-12    38-12
$ENVM    5-0#     9-19     9-68    38-5     38-5     38-12
$EOP    11-35    12-31    13-20#   37-20
$EOPCT   9-19*   13-20    13-20#
$EOSP   12-60    12-88    13-9#
$ERFLG   5-0#    38-6     38-6     38-6     38-6     38-6     38-6*    38-7     38-7     38-7*
$ERMAX   5-0#     9-19*   38-6     38-6     38-6     38-6*
$ERROR   9-19    38-7#
$ERRPC   5-0#    14-34    38-7     38-7     38-7     38-7*    38-7*
$ERRTB   7-0#    14-45
$ERTTL   5-0#    13-20    13-20*   38-7     38-7     38-7*
$ESCAP   5-0#     9-19*   38-6*    38-7     38-7     38-7     38-7
$ETABL   5-0#
$ETEND   4-789    5-0#
$FATAL   5-0#    38-12*
$FFLG   38-12    38-12#   38-12*   38-12*   38-12*
$FILLC   5-0#    38-5     38-5     38-5
$FILLS   5-0#    38-5     38-5
$GDADR   5-0#    12-534*  12-537   12-:06*  12-:09   18-83*   49-8
$GDDAT   5-0#    12-521*  12-522*  12-532*  12-533*  12-541   12-991*  12-992*  12-:04*  12-:05*  12-:13   12-B15*  12-B30*  12-B43*
        12-C08*  12-C23*  12-C36*  12-D84*  12-E46*  12-F69*  12-F70*  12-G28*  12-G29*  12-G83*  12-G84*  12-H38*  12-H39*  18-81*
        24-49*   24-50*   24-51    24-67*   24-68*   24-74*   24-76*   24-88*   24-89*   24-90*   24-106*  24-107*  24-122*  24-123*
        24-150*  24-151*  25-42*   25-55*   25-66*   25-67    25-79*   25-84*   25-87    25-98*   25-106*  25-109   25-135*  25-138*
        25-139*  25-142   25-154*  25-155*  25-158   25-165*  26-16*   26-23*   26-24*   26-25*   26-26*   26-30*   26-32    26-86*
        26-91*   26-94    26-104*  26-117*  26-122   26-134*  26-135*  26-136*  26-138   26-148*  26-149*  26-151   27-40*   27-55*
        29-36*   29-47*   29-63*   29-68    29-87*   29-101*  29-102*  29-129*  29-142*  29-166*  29-167*  29-181*  29-194*  29-195*
        29-210*  29-211*  31-31*   31-32    31-41*   31-57*   31-59    31-68*   31-81*   31-82    31-91*   31-103*  31-104   31-113*
        31-126*  31-127   32-25*   32-26*   32-39*   32-40*   32-58*   32-59*   32-72*   32-73*   32-86*   32-87*   32-100*  32-101*
        32-114*  32-115*  33-33*   33-34*   33-47*   33-48*   33-63*   33-64*   33-80*   33-81*   33-101*  33-102*  33-118*  33-119*
        33-132*  33-133*  33-157*  33-158*  33-173*  33-175*  33-187*  33-188*  33-203*  33-204*  33-220*  33-221*  33-238*  33-239*
        33-252*  33-253*  33-268*  33-269*  34-18*   34-19    34-30*   34-31    34-40*   34-71*   34-72    34-83*   34-84    34-92*
        35-29*   35-30*   35-45*   35-46*   35-65*   35-66*   35-85*   35-86*   35-114*  35-115*  35-133*  35-134*  35-147*  35-148*
        35-161*  35-162*  35-172*  35-187*  35-191   35-205*  35-206*  35-238*  35-239*  35-254*  35-255*  35-268*  35-269*  35-286*
        35-287*  35-299*  35-300*  35-314*  35-315*  35-323*  35-354*  35-355*  35-367*  35-368*  35-380*  35-381*  35-393*  35-394*
        35-415*  35-416*  35-440*  35-441*  35-455*  35-456*  35-468*  35-469*  35-481*  35-482*  35-494*  35-495*  35-515*  35-516*
        35-532*  35-533*  35-548*  35-549*  36-44*   36-45*   36-60*   36-76*   36-77*   36-96*   36-97*   36-110*  36-111*  49-1
        49-6     49-8     49-10    49-11
$GET42  13-20#
$GTSWR  38-8#    38-10    38-10
$HD      4-487    4-487    4-487
$HIBTS   4-789#
$HIOCT  38-9#    38-9*
$ICNT    5-0#    38-6     38-6     38-6     38-6*    38-6*
$ILLUP  38-11    38-11    38-11#
$INTAG   5-0#    38-8     38-8     38-8     38-8     38-8*
```

SITEM@	5-0#	14-29	38-7	38-7	38-7	38-7*

SEQ 0303

```
$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*   12-142* 12-188* 12-289* 12-335* 12-430* 12-476* 12-608* 12-655* 12-756* 12-803* 12-901* 12-947*
        12-:80*  12-;33* 12-<32* 12-<94* 12->17* 12->85* 12-?99* 12-@51* 12-A57* 12-B53* 12-D45* 12-E08* 12-F78* 12-H00*
        12-I07*  12-I57* 12-J62* 12-K15* 12-L14* 12-L60* 12-M59* 12-N05* 12-007* 12-053* 12-P54* 12-Q11* 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   12-238  12-380  12-555  12-703
        12-848   12-:27  12-;78  12-=63  12-?46  12-@96  12-C82  12-F07  12-G39  12-H57  12-J12  12-K63  12-M08  12-N53
        12-P04   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#  12-238  12-238#
        12-238#  12-380   12-380# 12-380# 12-555  12-555# 12-555# 12-703  12-703# 12-703# 12-848  12-848# 12-848# 12-:27
        12-:27#  12-:27#  12-;78  12-;78# 12-;78# 12-=63  12-=63# 12-=63# 12-?46  12-?46# 12-?46# 12-@96  12-@96# 12-@96#
        12-C82   12-C82#  12-C82# 12-F07  12-F07# 12-F07# 12-G39  12-G39# 12-G39# 12-H57  12-H57# 12-H57# 12-J12  12-J12#
        12-J12#  12-K63   12-K63# 12-K63# 12-M08  12-M08# 12-M08# 12-N53  12-N53# 12-N53# 12-P04  12-P04# 12-P04#
$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
```

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#
SEQ 0307

	38-10#	38-10#	38-10#	38-10#	38-10#	38-10#	38-10#	38-10#	38-10#	38-10#				
$TRPAD	38-10	38-10#												
$TSTM	4-789#													
$TSTNM	5-0#	11-37*	13-20*	38-6	38-6	38-6	38-6	38-6	38-6*	38-6*	38-7	38-7	38-7	
$TTYIN	38-8	38-8	38-8	38-8	38-8	38-8#								
$TYPBN	38-2#	38-10	38-10											
$TYPDS	38-3#	38-10	38-10											
$TYPE	38-5#	38-10	38-10	38-12										
$TYPEC	38-5	38-5	38-5	38-5#	38-8									
$TYPEX	38-5	38-5	38-5#											
$TYPOC	38-4#	38-10	38-10											
$TYPON	38-4	38-4#	38-10											
$TYPOS	38-4#	38-10												
$UNIT	5-0#	11-42*	14-23	24-49										
$UNITM	4-789#													
$USWR	5-0#													
$VECT1	5-0#	10-69	10-81*	10-85	10-107*									
$VECT2	5-0#													
$XOFF	38-5	38-5												
$XON	38-5	38-5												
$XTSTR	38-6#													
.$ASTA	38-12	38-12												
.$X	4-789	4-789#												
A16	4-728#													
A17	4-727#													
ABASE	4-779#	5-0	5-0											
ACDW1	5-0	5-0												
ACDW2	5-0	5-0												
ACKSTS	15-152	16-137	32-15#											
ACPUOP	5-0	5-0												
ADDW0	5-0	5-0												
ADDW1	5-0	5-0												
ADDW10	5-0													
ADDW11	5-0													
ADDW12	5-0													
ADDW13	5-0													
ADDW14	5-0													
ADDW15	5-0													
ADDW2	5-0	5-0												
ADDW3	5-0	5-0												
ADDW4	5-0	5-0												
ADDW5	5-0	5-0												
ADDW6	5-0	5-0												
ADDW7	5-0	5-0												
ADDW8	5-0													
ADDW9	5-0													
ADEVCT	5-0	5-0												
ADEVM	5-0	5-0												
ADR	12-2	12-2#	12-2#	12-34	12-34#	12-34#	12-63	12-63#	12-63#	12-91	12-91#	12-91#	12-238	12-238#
	12-238#	12-380	12-380#	12-380#	12-555	12-555#	12-555#	12-703	12-703#	12-703#	12-848	12-848#	12-848#	12-:27
	12-:27#	12-:27#	12-;78	12-;78#	12-;78#	12-=63	12-=63#	12-=63#	12-?46	12-?46#	12-?46#	12-a96	12-a96#	12-a96#
	12-C82	12-C82#	12-C82#	12-F07	12-F07#	12-F07#	12-G39	12-G39#	12-G39#	12-H57	12-H57#	12-H57#	12-J12	12-J12#
	12-J12#	12-K63	12-K63#	12-K63#	12-M08	12-M08#	12-M08#	12-N53	12-N53#	12-N53#	12-P04	12-P04#	12-P04#	
AENV	5-0	5-0												
AENVM	5-0	5-0												
AFATAL	5-0	5-0												

ALL 10-12¹ 39-6#

SEQ 0309

CROSS REFERENCE TABLE (CREF V01-05) SEQ 0310

B 9

12-L52# 12-L54# 12-L56# 12-L69 12-L69# 12-L85# 12-L88# 12-L90# 12-L93# 12-L97# 12-L99# 12-M01# 12-M03# 12-M16

SEQ 0311

12-A06# 12-A17 12-A17# 12-A41 12-A41# 12-A46# 12-A46
SEQ 0313

12-A89 12-A89# 12-A89# 12-A91 12-A91# 12-A91# 12-A06

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

F 9

SEQ 0315

39-4 39-6 39-9 39-10 39-11 39-12 39-12H9 39-13 39-15 39-16 39-17 39-18 39-20 39-22

SEQ 0317

	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											
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-F32	12-F91										
DPE	4-688#	24-137	25-255	29-32	33-31	35-43	35-466	35-469						
DPEHI	4-758#													
DPELO	4-759#													
DPR	4-566#	29-195	31-125	31-126	34-30									
DRQ	4-650#													
DRVCLR	4-511#	16-87	34-18											
DRVSTS	16-105	34-9#												
DRY	4-567#	24-104	25-38	25-41	25-42	29-195	31-125	31-126	34-30					
DSWR	4-491#	5-0	9-19											
DTASTS	12-138	12-182	12-227	12-285	12-329	12-373	12-426	12-470	12-519	12-604	12-649	12-694	12-752	12-797
	12-841	12-897	12-941	12-989	12-:76	12-;27	12-;71	12-<28	12-<88	12-=49	12->13	12->79	12-?39	12-?95
	12-@45	12-@89	12-A53	12-B12	12-B28	12-B40	12-C05	12-C21	12-C33	12-D41	12-E02	12-E64	12-F66	12-G26
	12-I03	12-I49	12-I98	12-J58	12-K07	12-K54	12-L10	12-L54	12-L99	12-M55	12-M99	12-N44	12-003	12-047
	12-092	12-P50	12-005	12-066	16-200	35-17#								
DTE	4-576#	4-590	35-279	35-297	35-300									
DTO	4-614#													
DULPRT	4-653#	12-71	12-76	12-81	12-85									
DVA	4-497#	9-89	12-51	20-43	21-37	24-65	24-68	28-37	31-31	34-17	34-18			
DVC	4-687#	29-127	31-115	33-92	33-130	33-133	33-266	35-232	35-236	35-239	35-284	36-89	36-94	36-97
EARLY	12-K84	42-107#												
EBL	4-630#													
ECH	4-582#	4-590	16-225	18-25	25-452	25-467	25-485	25-491	35-423	40-77	40-78	40-85	40-86	
ECI	4-658#	18-21	25-487	35-420										
ECRC	4-634#													
ED1	45-1	49-1#												
ED110	45-4	49-2#												
ED111	45-5	49-3#												
ED114	45-7	49-5#												
ED223	45-8	49-6#												
ED336	45-10	49-8#												
ED337	45-11	45-12	49-10#											
ED353	45-14	49-11#												
EDT1	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 J 9 7-303 7-306 7-309 7-312 7-315 7-318 7-321

SEQ 0319

CROSS REFERENCE TABLE (CREF V01-05)

SEQ 0320

	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	7-376	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	7-433	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
	45-1#													
EDT110	7-222	45-4#												
EDT111	7-225	7-228	45-5#											
EDT114	7-234	45-7#												
EDT2	7-427	7-430	7-514	7-517	45-2#									
EDT223	7-448	7-529	45-8#											
EDT336	7-352	7-682	7-688	7-691	45-10#									
EDT337	7-685	45-11#												
EDT344	7-700	45-12#												
EDT353	7-721	45-14#												
EECC	4-639#													
EF110	46-4	50-1#												
EF111	46-1	46-5	50-2#											
EF114	46-7	46-8	46-14	50-3#										
EF336	46-10	46-11	46-12	50-4#										
EF337	50-5#													
EFT1	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	7-376	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	7-433	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
	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#

L 9

SEQ 0321

CROSS REFERENCE TABLE (CREF V01-05)

SEQ 0322

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	7-376	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	7-433	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#												
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#

N 9

SEQ 0323

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	43-19?	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#												
EMS162	47-116#													
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#

SEQ 0325

E 10

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

SEQ 0327

	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#												
EMT110	7-222	43-74#												
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#

G 10

SEQ 0329

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#
EMT173	7-376	43-125#
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- 00	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#

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#
EMT256	7-529	43-176#
EMT257	7-532	43-177#
EMT26	7-67	43-24#
EMT260	7-535	43-178#
EMT261	7-538	43-179#
EMT262	7-541	43-180#
EMT263	7-544	43-181#
EMT264	7-547	43-182#
EMT265	7-550	43-183#
EMT266	7-553	43-184#
EMT267	7-557	43-185#
EMT27	7-70	43-25#
EMT270	7-560	43-186#
EMT271	7-564	43-187#
EMT272	7-567	43-188#
EMT273	7-570	43-189#
EMT274	7-574	43-190#
EMT275	7-578	43-191#
EMT276	7-582	43-192#
EMT277	7-587	43-193#
EMT3	7-10	43-5#
EMT30	7-73	43-26#

EMT301	7-595	43-195#	
EMT302	7-598	43-196#	
EMT303	7-601	43-197#	
EMT304	7-604	43-198#	
EMT305	7-607	43-199#	
EMT306	7-610	43-200#	
EMT307	7-613	43-201#	
EMT31	7-76	43-27#	
EMT310	7-616	43-202#	
EMT311	7-619	43-203#	
EMT312	7-622	43-204#	
EMT313	7-625	43-205#	
EMT314	7-628	43-206#	
EMT315	7-631	43-207#	
EMT316	7-634	43-208#	
EMT317	7-637	43-209#	
EMT32	7-79	43-28#	
EMT320	7-640	43-210#	
EMT321	7-643	43-211#	
EMT322	7-646	43-212#	
EMT323	7-649	43-213#	
EMT324	7-652	43-214#	
EMT325	7-655	43-215#	
EMT326	7-658	43-216#	
EMT327	7-661	43-217#	
EMT33	7-82	43-29#	
EMT330	7-664	43-218#	
EMT331	7-667	43-219#	
EMT332	7-670	43-220#	
EMT333	7-673	43-221#	
EMT334	7-676	43-222#	
EMT335	7-679	43-223#	
EMT336	7-352	7-682	43-224#
EMT337	7-685	43-225#	
EMT34	7-85	43-30#	
EMT340	7-688	43-226#	
EMT341	7-691	43-227#	
EMT342	7-694	43-228#	
EMT343	7-697	43-229#	
EMT344	7-700	43-230#	
EMT345	7-703	43-231#	
EMT346	7-706	43-232#	
EMT347	7-709	43-233#	
EMT35	7-88	43-31#	
EMT350	7-712	43-234#	
EMT351	7-715	43-235#	
EMT352	7-718	43-236#	
EMT353	7-721	43-237#	
EMT354	7-724	43-238#	
EMT36	7-91	43-32#	
EMT37	7-94	43-33#	
EMT4	7-13	43-6#	
EMT40	7-97	43-34#	
EMT41	7-100	43-35#	
EMT42	7-103	43-36#	
EMT43	7-106	43-37#	

EMT44 7-109 43-38#

M 10

SEQ 0335

12-134# 12-134# 12-138 12-138# 12-138# 12-151 12-151# 12-151# 12-168 12-168# 12-168# 12-171 12-171# 12-173

D 11

12-E33 12-E33# 12-E33# 12-E36 12-E36# 12-E38 12-E38# 12-E41 12-E41# 12-E41# 12-E45 12-E45# 12-E45# 12-E64

SEQ 0339

12-08¹ 12-P41 12-P94 12-Q55 15-46 16-75 19-10# F 11

SEQ 0341

OR 4-742#

J 11

SEQ 0345

RGDTP[T] 42-3#

L 11

SEQ 0347

EGGGM[illegible] N P

RMDBO 6-0#

N 11

SEQ 0349

CC

P

PRRSSSSSS

STTTTTTTTTX

33-119 33-218 35-173 35-176 35-193 35-204 35-206 35-207 35-209 35-221 35-513 36-74 36-89 36-108

TYPOS	9-42	9-53	9-99	10-70	10-91	14-23	14-27	14-32	14-34	38-10#				
U0	4-749#													
U1	4-749#													
U2	4-749#													
UBUSQS	10-42	39-14#												
UNS	4-574#	12-G58	12-H13	31-70	32-51	32-55	32-59	33-232	33-264	33-269	35-279	35-282	35-287	
UNTMSK	4-753#	24-48	24-50											
UPE	4-736#	25-270												
USE	4-693#	12-B04	12-B97	17-29										
USRFIL	16-254	16-257	16-267*	16-275*	16-326	51-12#								
VV	4-568#	15-122	16-116	25-196	29-91	29-156	29-157	29-206	29-211	32-23	32-26	33-106	33-145	33-146
	33-169	33-175	34-29	35-90	35-119	35-505	35-506	35-544	35-549	36-35	36-36	36-56	36-61	
WC	4-638#	31-80	34-70											
WCD	4-524#	12-349	12-492	12-817	12-963	12-;47	12-?15	12-@65	12-K30	12-Q47				
WCE	4-735#	12-516	12-522	12-987	12-992	25-169	40-77	40-78						
WCEHI	4-760#													
WCELO	4-761#													
WCF	4-583#	4-590	25-240	35-479	35-482									
WCH	4-525#													
WD	4-528#	12-158	12-305	12-446	12-625	12-773	12-917	12-:85	12-;03	12-<66	12->24	12->55	12-@02	12-@21
	12-A73	12-D60	12-F19	12-F37	12-G50	12-G63	12-I25	12-J80	12-L30	12-M75	12-Q23			
WH	4-529#	12-108	12-255	12-397	12-572	12-720	12-865	12-:44	12-;96	12-=76	12-?59	12-A15	12-C62	12-C67
	12-D01	12-E87	12-E92	12-H73	12-J28	12-K80	12-M25	12-N71	12-P20	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	12-112	12-259	12-401	12-:48	12->22	12-?65	12-A19	12-H77	12-P24	42-38#				

```
$$CMRE   4-791#
$$CMTM   4-791#  5-0      5-0      5-0      5-0      5-0
$$ESCA   4-491#
$$NEWT   4-491#  12-2     12-34    12-63    12-91    12-238   12-380   12-555   12-703   12-848   12-:27   12-;78   12-=63   12-?46
        12-@96   12-C82   12-F07   12-G39   12-H57   12-J12   12-K63   12-M08   12-N53   12-P04
$$SET    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10    38-10#
$$SETM   9-19     9-19#
$$SKIP   4-491#
.$ACT1   4-483#   4-786
.$APTB   4-483#   5-0      5-0#
.$APTH   4-483#   4-789
.$APTY   4-483#   38-12
.$CATC   4-479#   4-784
.$CMTA   4-480#   4-791
.$EOP    4-480#   13-20
.$ERRO   4-480#   38-7
.$ERRT   4-480#
.$POWE   4-482#   38-11
.$RDDE   4-481#
.$RDOC   4-481#   38-9
.$READ   4-481#   38-8
.$SAVE   4-482#   38-1
.$SCOP   4-480#   38-6
.$SIZE   4-482#
.$TRAP   4-482#   38-10
.$TYPB   4-481#   38-2
.$TYPD   4-481#   38-3
.$TYPE   4-480#   38-5
.$TYPO   4-481#   38-4
.EQUAT   4-479#   4-491
.HEADE   4-479#   4-487
.SETUP   4-479#   4-782
.SWRHI   4-479#   4-488
.SWRLO   4-479#   4-488#   4-489
CALCLR   4-194#   12-36    12-65
CALSUB   4-208#   12-99    12-110   12-126   12-129   12-131   12-134   12-138   12-151   12-168   12-171   12-173   12-176   12-180
        12-182   12-184   12-197   12-213   12-216   12-218   12-221   12-225   12-227   12-229   12-233   12-246   12-257   12-273
        12-276   12-278   12-281   12-285   12-298   12-315   12-318   12-320   12-323   12-327   12-329   12-331   12-344   12-359
        12-362   12-364   12-367   12-371   12-373   12-375   12-388   12-399   12-414   12-417   12-419   12-422   12-426   12-439
        12-456   12-459   12-461   12-464   12-468   12-470   12-472   12-487   12-502   12-505   12-507   12-510   12-514   12-519
        12-547   12-563   12-574   12-592   12-595   12-597   12-600   12-604   12-618   12-635   12-638   12-640   12-643   12-647
        12-649   12-651   12-664   12-680   12-683   12-685   12-688   12-692   12-694   12-696   12-698   12-711   12-722   12-740
        12-743   12-745   12-748   12-752   12-766   12-783   12-786   12-788   12-791   12-795   12-797   12-799   12-812   12-827
        12-830   12-832   12-835   12-839   12-841   12-843   12-856   12-867   12-885   12-888   12-890   12-893   12-897   12-910
        12-927   12-930   12-932   12-935   12-939   12-941   12-943   12-958   12-973   12-976   12-978   12-981   12-985   12-989
        12-:19   12-:35   12-:46   12-:64   12-:67   12-:69   12-:72   12-:76   12-:98   12-;13   12-;16   12-;18   12-;21   12-;25
        12-;27   12-;29   12-;42   12-;57   12-;60   12-;62   12-;65   12-;69   12-;71   12-;73   12-;86   12-;98   12-<16   12-<19
        12-<21   12-<24   12-<28   12-<41   12-<46   12-<49   12-<51   12-<54   12-<58   12-<76   12-<79   12-<82   12-<86   12-<88
        12-<90   12-=03   12-=08   12-=11   12-=13   12-=16   12-=20   12-=37   12-=40   12-=43   12-=47   12-=49   12-=51   12-=53
        12-=78   12-=87   12->01   12->04   12->06   12->09   12->13   12->34   12->38   12->41   12->43   12->46   12->50   12->65
        12->68   12->70   12->73   12->77   12->79   12->81   12->94   12->98   12-?01   12-?03   12-?06   12-?10   12-?25   12-?28
        12-?30   12-?33   12-?37   12-?39   12-?41   12-?61   12-?70   12-?83   12-?86   12-?88   12-?91   12-?95   12-@16   12-@31
        12-@34   12-@36   12-@39   12-@43   12-@45   12-@47   12-@60   12-@75   12-@78   12-@80   12-@83   12-@87   12-@89   12-@91
        12-A06   12-A17   12-A41   12-A44   12-A46   12-A49   12-A53   12-A66   12-A83   12-A86   12-A88   12-A91   12-A95   12-B12
        12-B28   12-B40   12-B49   12-B61   12-B78   12-B81   12-B83   12-B86   12-B90   12-C05   12-C21   12-C33   12-C42   12-C65
        12-C76   12-C92   12-D03   12-D29   12-D32   12-D34   12-D37   12-D41   12-D54   12-D71   12-D74   12-D76   12-D79   12-D83
```

```
                12-E02  12-E04  12-E17  12-E33  12-E36  12-E38  12-E41  12-E45  12-E64  12-E66  12-E68  12-E90  12-F01  12-F21
                12-F28  12-F49  12-F52  12-F54  12-F57  12-F61  12-F66  12-F74  12-F87  12-G09  12-G12  12-G14  12-G17  12-G21
                12-G26  12-G34  12-G54  12-G73  12-G76  12-G79  12-G89  12-G92  12-G94  12-G96  12-H09  12-H28  12-H31  12-H34
                12-H45  12-H48  12-H50  12-H52  12-H64  12-H75  12-H91  12-H94  12-H96  12-H99  12-I03  12-I17  12-I35  12-I38
                12-I40  12-I43  12-I47  12-I49  12-I53  12-I66  12-I84  12-I87  12-I89  12-I92  12-I96  12-I98  12-J02  12-J04
                12-J18  12-J30  12-J46  12-J49  12-J51  12-J54  12-J58  12-J72  12-J93  12-J96  12-J98  12-K01  12-K05  12-K07
                12-K11  12-K24  12-K40  12-K43  12-K45  12-K48  12-K52  12-K54  12-K58  12-K71  12-K82  12-K98  12-L01  12-L03
                12-L06  12-L10  12-L23  12-L40  12-L43  12-L45  12-L48  12-L52  12-L54  12-L56  12-L69  12-L85  12-L88  12-L90
                12-L93  12-L97  12-L99  12-M01  12-M03  12-M16  12-M27  12-M43  12-M46  12-M48  12-M51  12-M55  12-M68  12-M85
                12-M88  12-M90  12-M93  12-M97  12-M99  12-N01  12-N14  12-N30  12-N33  12-N35  12-N38  12-N42  12-N44  12-N46
                12-N48  12-N61  12-N73  12-N91  12-N94  12-N96  12-N99  12-003  12-016  12-033  12-036  12-038  12-041  12-045
                12-047  12-049  12-062  12-078  12-081  12-083  12-086  12-090  12-092  12-094  12-096  12-P11  12-P22  12-P38
                12-P41  12-P43  12-P46  12-P50  12-P63  12-P75  12-P81  12-P84  12-P91  12-P94  12-P96  12-P99  12-Q03  12-Q05
                12-Q07  12-Q19  12-Q32  12-Q41  12-Q44  12-Q52  12-Q55  12-Q57  12-Q60  12-Q64  12-Q66  12-Q68
COMMEN           4-491#
ENDCOM           4-491#
ERROR            4-12#  11-23   12-26   12-36   12-49   12-53   12-59   12-65   12-67   12-86   12-99   12-110  12-126  12-134
                12-138  12-151  12-168  12-176  12-180  12-182  12-184  12-197  12-213  12-221  12-225  12-227  12-229  12-233
                12-246  12-257  12-273  12-281  12-285  12-298  12-315  12-323  12-327  12-329  12-331  12-344  12-359  12-367
                12-371  12-373  12-375  12-388  12-399  12-414  12-422  12-426  12-439  12-456  12-464  12-468  12-470  12-472
                12-487  12-502  12-510  12-514  12-519  12-526  12-529  12-539  12-543  12-547  12-563  12-574  12-592  12-600
                12-604  12-618  12-635  12-643  12-647  12-649  12-651  12-664  12-680  12-688  12-692  12-694  12-696  12-698
                12-711  12-722  12-740  12-748  12-752  12-766  12-783  12-791  12-795  12-797  12-799  12-812  12-827  12-835
                12-839  12-841  12-843  12-856  12-867  12-885  12-893  12-897  12-910  12-927  12-935  12-939  12-941  12-943
                12-958  12-973  12-981  12-985  12-989  12-996  12-:00  12-:11  12-:15  12-:19  12-:35  12-:46  12-:64  12-:72
                12-:76  12-:98  12-;13  12-;21  12-;25  12-;27  12-;29  12-;42  12-;57  12-;65  12-;69  12-;71  12-;73  12-;86
                12-;98  12-<16  12-<24  12-<28  12-<41  12-<46  12-<54  12-<58  12-<76  12-<82  12-<86  12-<88  12-<90  12-=03
                12-=08  12-=16  12-=20  12-=37  12-=43  12-=47  12-=49  12-=51  12-=53  12-=78  12-=87  12->01  12->09  12->13
                12->34  12->38  12->46  12->50  12->65  12->73  12->77  12->79  12->81  12->94  12->98  12-?06  12-?10  12-?25
                12-?33  12-?37  12-?39  12-?41  12-?61  12-?70  12-?83  12-?91  12-?95  12-@16  12-@31  12-@39  12-@43  12-@45
                12-@47  12-@60  12-@75  12-@83  12-@87  12-@89  12-@91  12-A06  12-A17  12-A41  12-A49  12-A53  12-A66  12-A83
                12-A91  12-A95  12-B12  12-B20  12-B28  12-B32  12-B40  12-B44  12-B49  12-B61  12-B78  12-B86  12-B90  12-C05
                12-C13  12-C21  12-C25  12-C33  12-C37  12-C42  12-C65  12-C76  12-C92  12-D03  12-D29  12-D37  12-D41  12-D54
                12-D71  12-D79  12-D83  12-D97  12-E02  12-E04  12-E17  12-E33  12-E41  12-E45  12-E59  12-E64  12-E66  12-E68
                12-E90  12-F01  12-F21  12-F28  12-F32  12-F49  12-F57  12-F61  12-F66  12-F71  12-F74  12-F87  12-F91  12-G09
                12-G17  12-G21  12-G26  12-G31  12-G34  12-G54  12-G58  12-G73  12-G79  12-G86  12-G92  12-G94  12-G96  12-H09
                12-H13  12-H28  12-H34  12-H41  12-H48  12-H50  12-H52  12-H64  12-H75  12-H91  12-H99  12-I03  12-I17  12-I35
                12-I43  12-I47  12-I49  12-I53  12-I66  12-I84  12-I92  12-I96  12-I98  12-J02  12-J04  12-J18  12-J30  12-J46
                12-J54  12-J58  12-J72  12-J93  12-K01  12-K05  12-K07  12-K11  12-K24  12-K40  12-K48  12-K52  12-K54  12-K58
                12-K71  12-K82  12-K98  12-L06  12-L10  12-L23  12-L40  12-L48  12-L52  12-L54  12-L56  12-L69  12-L85  12-L93
                12-L97  12-L99  12-M01  12-M03  12-M16  12-M27  12-M43  12-M51  12-M55  12-M68  12-M85  12-M93  12-M97  12-M99
                12-N01  12-N14  12-N30  12-N38  12-N42  12-N44  12-N46  12-N48  12-N61  12-N73  12-N91  12-N99  12-003  12-016
                12-033  12-041  12-045  12-047  12-049  12-062  12-078  12-086  12-090  12-092  12-094  12-096  12-P11  12-P22
                12-P38  12-P46  12-P50  12-P63  12-P75  12-P81  12-P91  12-P99  12-Q03  12-Q05  12-Q07  12-Q19  12-Q32  12-Q41
                12-Q52  12-Q60  12-Q64  12-Q66  12-Q68
ESCAPE           4-491#
GETPRI           4-491#
GETREG           4-171#  12-67   12-529  12-:00
GETSWR           4-491#   9-25    9-25#  11-27
MSG              4-8#
MULT             4-491#
NEWTST           4-491#  12-2    12-34   12-63   12-91   12-238  12-380  12-555  12-703  12-848  12-:27  12-;78  12-=63  12-?46
                12-@96  12-C82  12-F07  12-G39  12-H57  12-J12  12-K63  12-M08  12-N53  12-P04
NWTST            4-44#  12-2    12-34   12-63   12-91   12-238  12-380  12-555  12-703  12-848  12-:27  12-;78  12-=63  12-?46
                12-@96  12-C82  12-F07  12-G39  12-H57  12-J12  12-K63  12-M08  12-N53  12-P04
POP              4-491#  12-19   12-20   12-24   12-25   12-45   12-46   12-57   12-58   16-71   16-279  16-289  16-399  17-65
```

18-95	19-25	19-26	19-27	20-70	21-59	22-20 M 12	22-21	23-35	24-145	24-149	25-131	28-51	30-25 SEQ 0361