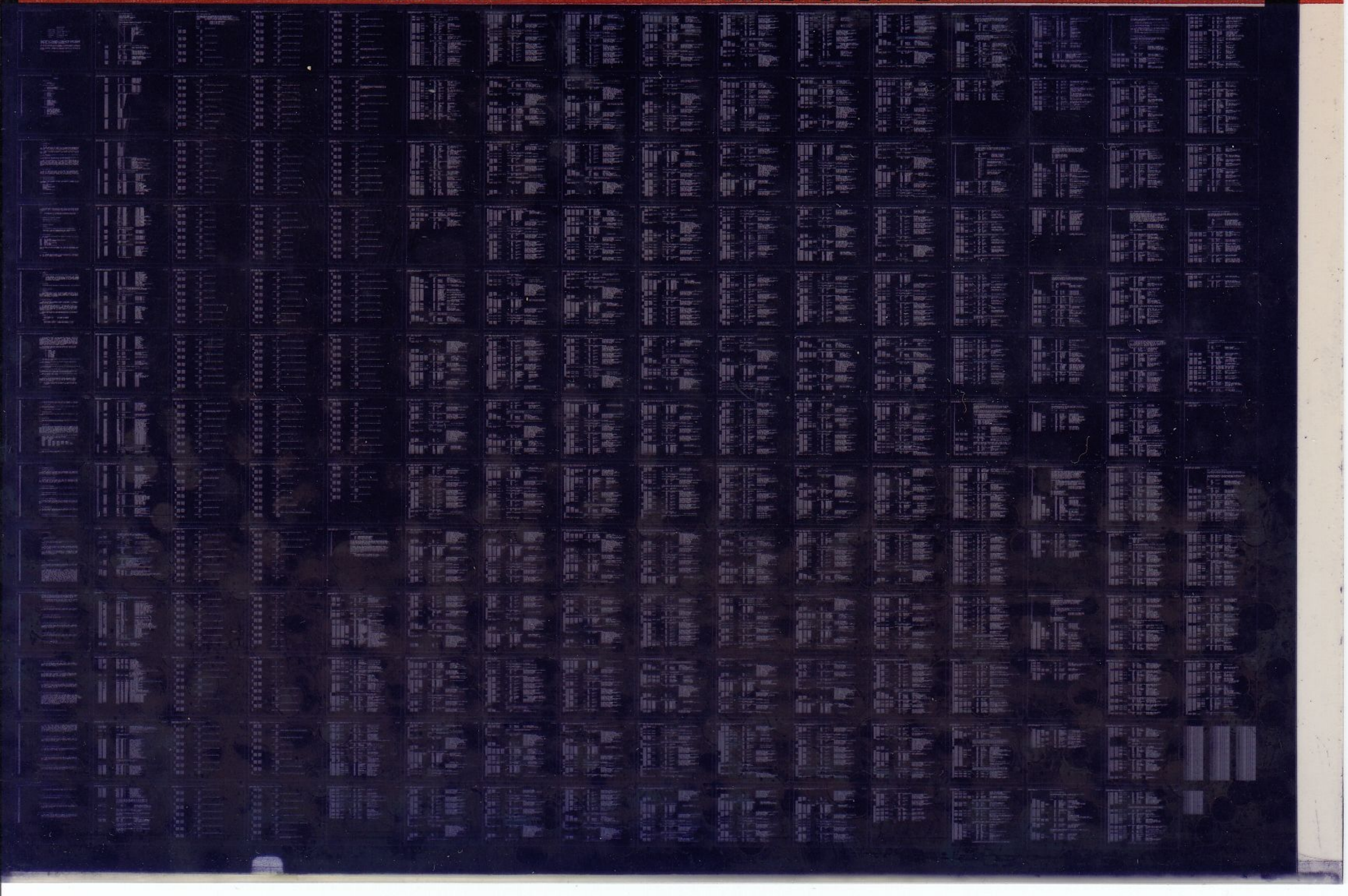
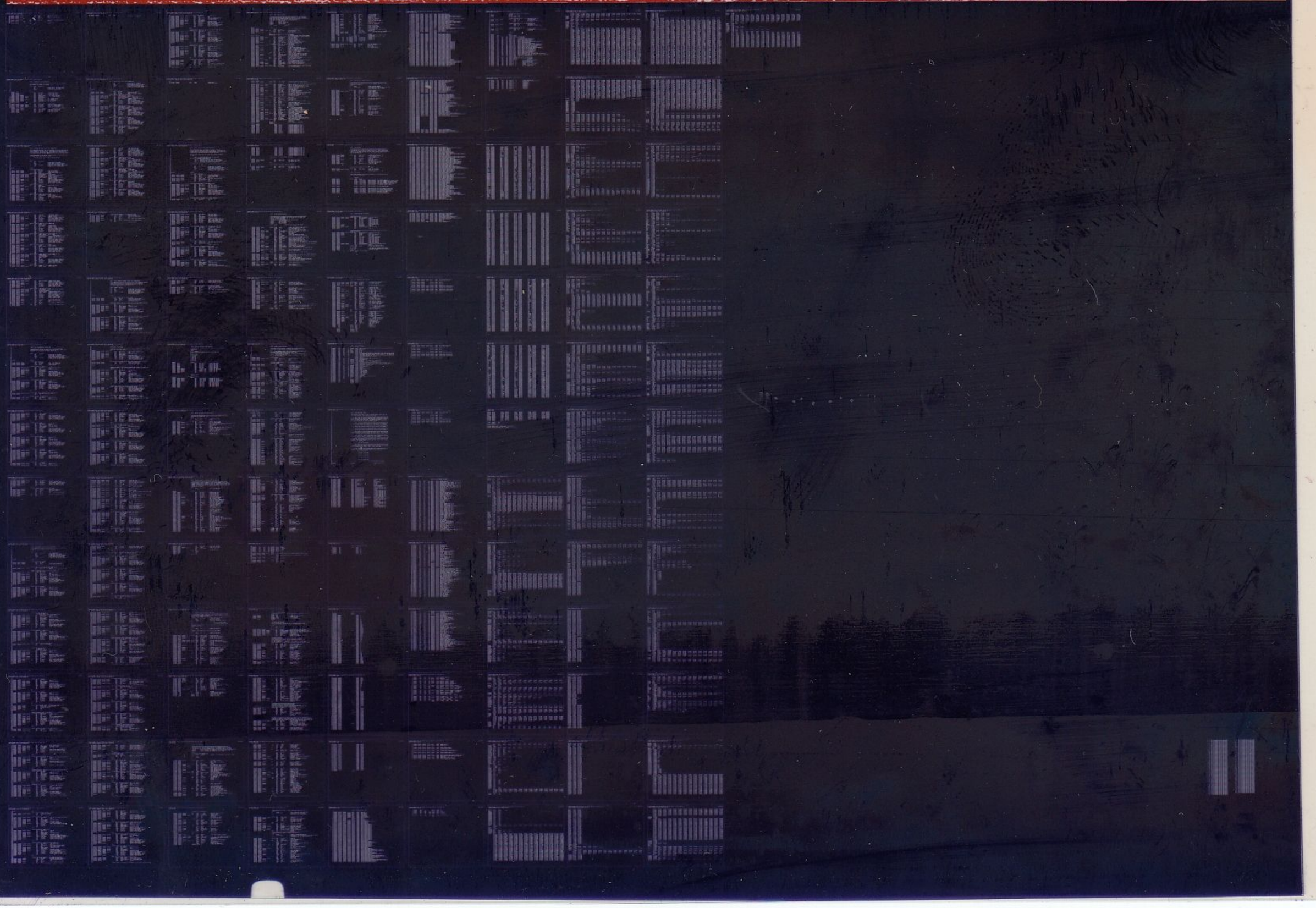
RM80 FCTNL TEST PT 3 CZRNFA0 AH-T115A-MC FICHE 1 OF 2



RM80

FCTNL TEST PT 3
CZRNFA0

AH-T115A-MC
FICHE 2 OF 2



```
.REM    \
```

IDENTIFICATION

PRODUCT CODE: AC-T114A-MC

PRODUCT NAME: CZRNFA0 RM80 FCTNL PT 3

PRODUCT DATE: APRIL 1, 1982

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

CONTENTS

1.0 INTRODUCTION

1.1 ABSTRACT

THE RM80 SUBSYSTEM FUNCTIONAL TEST IS A STAND ALONE PROGRAM WHICH USES FUNCTIONAL MEANS TO VERIFY THE OPERABILITY OF THE RM80 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 RM80 SUBSYSTEM, INCLUDING THE MASSBUS CONTROLLER, MASSBUS ADAPTER AND THE DISK DRIVE.

THE TEST IS COMPRISED OF 4 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; PART 4 TESTS MECHANICAL POSITIONING OPERATIONS AND VARIOUS TIMING PARAMETERS OF THE RM80 DISK DRIVE.

1.2 UNIT UNDER TEST

THE UNIT UNDER TEST (UUT) IS THE RM80 DISK SUBSYSTEM WHICH CONSISTS OF THE RH70 MASSBUS CONTROLLER ,THE RM80 MASSBUS ADAPTER , AND THE STORAGE MODULE DISK DRIVE. NOTE THAT A DISK 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 RM80 SUBSYSTEM FUNCTIONAL TEST:

PDP-11 PROCESSOR
20K MEMORY
KW11-L OR KW11-P CLOCK
PROGRAM LOADING DEVICE
TERMINAL
RH70 CONTROLLER
1 TO 8 RM80 DISK DRIVES

2.2 MEDIA REQUIREMENTS

EACH UNIT BEING TESTED MUST BE LOADED WITH A DISK BEFORE TESTING BEGINS ON THAT UNIT. THE DRIVE MUST HAVE A FORMATTED DISK AND CONTAIN A READABLE COPY OF THE MFG(DEC144), USR(DEC144) AND SSF BAD SECTOR FILES.

NOTE: TO ENSURE THAT ALL CUSTOMER DATA IS PRESERVED ON THE DISK, ALL DATA TRANSFERS WILL BE PERFORMED ON THE FE CYLINDERS ONLY.

2.3 PREREQUISITE DIAGNOSTIC PROGRAMS

RM80 DISKLESS TEST, PART 1 & 2

RM80 FUNCTIONAL TEST, PART 1 & 2

3.0 OPERATING PROCEDURE

3.1 LOADING

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

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

3.2 SWITCH OPTIONS

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

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

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

3.3 STARTING

THE PROGRAM MAY BE STARTED AT LOCATION 200 OR 204. STARTING AT 200 WILL BE THE NORMAL STARTING ADDRESS. STARTING AT 204 WILL ENABLE THE RH/RM BASE ADDRESS TO BE CHANGED.

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. THIS OF COURSE, DEPENDS ON WHICH TEST WAS BEING PERFORMED AT THE TIME THE PROGRAM WAS HALTED. ALSO, THIS ONLY APPLIES TO A NORMAL (ERROR FREE) OPERATION OF THE PROGRAM.

3.5 RESTARTING

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

4.0 OPERATOR INTERFACE

4.1 PROGRAM ID

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

4.2 CONSOLE DIALOGUE

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

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

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

EXAMPLE 1

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

EXAMPLE 2

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

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

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

```
'UNIT STATUS:
  0    ONLINE   RM80
  1    LOAD DEVICE
  2    OFFLINE  RM80
  3    NOT PRESENT
  4    NOT PRESENT
  5    NOT AN RM80
  6    NOT PRESENT
  7    NOT PRESENT'
```

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

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

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

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

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

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

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

```
'DRIVE 0'
```

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

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

4.3 PROGRESS REPORTS

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

'END OF PASS 1'

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

'TOTAL ERRORS SINCE LAST REPORT 0'

4.4 PERFORMANCE REPORT

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

4.5 PROGRAM HALTS

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

4.6 ERROR REPORTS

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

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

```
DRV# 0 - RM80, TEST# 14, ERR# 326, PC=016654
MASSBUS DATA BUS PARITY ERROR 'MDPE' (RMCS2, BIT 8) DETECTED
DURING WRITE COMMAND
EXPECTD    RECEVD
040300     040700
RMCS1      RMCS2   RMDS    RMER1   RMER2   RMAS
144252     040700  010700  000000  000000  000000
RMWC       RMBA    RMDA    RMOF    RMDC    RMEC1   RMEC2
177403     104604  000002  010000  000000  004066  000000
RMMR1      RMMR2   RMDT    RMSN
000010     011777  024026  177777
```

4.7 EXECUTION TIME

PASS 1 OF THE PROGRAM TAKES ABOUT 20 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE 50 SECONDS.

5.0 ENVIRONMENTAL SUPPORT

5.1 PROCESSOR COMPATIBILITY

THE RM80 SUBSYSTEM FUNCTIONAL TEST IS EXECUTABLE ON A PDP-11/70 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 RM80 SUBSYSTEM FUNCTIONAL TEST DOES NOT SPECIFICALLY TEST DUAL PORT LOGIC IN THE RM80 ADAPTER BUT IS EXECUTABLE ON RM80 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 RM80 SUSBYSTEM FUNCTIONAL TEST.

5.4 MEMORY MANAGEMENT HARDWARE

MEMORY MANAGEMENT HARDWARE IS NOT USED DURING THE RM80 SUSBYSTEM FUNCTIONAL TEST.

5.5 ACT11, APT11 COMPATIBILITY

THE RM80 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 RM80 SUSBYSTEM FUNCTIONAL TEST IS COMPATIBLE WITH XXDP IN DUMP AND CHAIN MODES, AND PROVIDES MEDIA PROTECTION IN THE CASE WHERE THE RM80 IS THE XXDP LOADING DEVICE.

5.7 OPERATING SYSTEM COMPATIBILITY

THE PROGRAM IS NOT COMPATIBLE WITH ANY SOFTWARE OPERATING SYSTEM.

6.0 TEST DESCRIPTION

TEST 1 CONTROLLER ACCESS TEST

PURPOSE:

TO VERIFY THAT THE UNIBUS ADDRESS OF THE RM80 SUBSYSTEM IS CORRECT, AS DEFINED AT LOCATION $BASE.

PROCEDURE:

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

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

TEST 2 - 22 WRITE/READ DATA TESTS

PURPOSE:

TO TEST WRITE DATA AND READ DATA FUNCTIONALITY OF THE RM80 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

```
400                         MODIFIED FOLLOWING THE SUCCESSFUL COMPLETION OF THE WRITE COMMAND
401                         IN  ORDER  TO  SHORTEN EXECUTION TIMES AND ENHANCE SCOPING LOOPS,
402                         THEN THE PROGRAM EXECUTES THE READ  DATA  PORTION  OF  THE  TEST,
403                         VERIFYING THE SAME TYPE OF ERRORS AS IN THE WRITE COMMAND.
404
405                         NOTE: THE SECTOR USED DURING A TEST MAY  DIFFER FROM THE  PROGRAM
406                               LISTING BECAUSE THE  PROGRAM SUBSTITUTES A GOOD SECTOR,  IF
407                               THE ONE SELECTED IS LISTED IN THE BAD BLOCK TABLE.
408
409
410
411
412
413                 TEST 2  WRITE, READ ZEROS TEST
414
415                      THE TEST WRITES AND READS AN ALL ZEROS DATA  FIELD,  CAUSING
416                 THE DRIVE TO USE NORMAL WRITE GATE THOUGHOUT THE WRITE PROCESS.
417
418
419
420
421
422                 TEST 3  WRITE, WRITE CHECK ZEROS TEST
423
424                      THE TEST WRITES AND WRITE CHECKS AN ALL  ZEROS  DATA  FIELD,
425                 VERIFYING THAT THERE ARE NO ERRORS.
426
427
428
429
430
431                 TEST 4  WRITE, WRITE CHECK ZEROS W/ WCE ERROR TEST
432
433                      THE TEST WRITES AN ALL ZEROS DATA  FIELD,  THEN  COMPLEMENTS
434                 EACH BIT IN THE LAST  WORD  OF  THE  WRITE BUFFER.  A WRITE CHECK
435                 COMMAND IS  EXECUTED AND THE TEST VERIFIES THAT THE CORRECT WRITE
436                 CHECK ERROR IS DETECTED.
437
438
439
440
441
442                 TEST 5  WRITE, READ ONES TEST
443
444                      THE TEST WRITES AND READS AN ALL ONES  DATA  FIELD,  CAUSING
445                 THE DRIVE TO USE NORMAL WRITE GATE THOUGHOUT THE WRITE PROCESS.
446
447
448
449
450
451                 TEST 6  WRITE, WRITE CHECK ONES TEST
452
453                      THE TEST WRITES AND WRITE CHECKS AN ALL ONES DATA FIELD.
454
455
456
```

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

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

TEST 10 WRITE, WRITE CHECK MULTIPLE SECTORS TEST

THE TEST SEEKS TO CYLINDER 559. 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. THE TEST IS REPEATED FOR 16 BIT FORMAT WITH SSEI BIT SET.

TEST 11 WRITE, READ W/ IMPLIED SEEK TEST

THIS TEST SEEKS TO THE FIRST CYLINDER PRIOR TO WRITING DATA ON CYLINDER 559., TRACK 2 AND SECTOR 0. THE EXPLICIT SEEK INSURES THAT THERE WILL BE MAXIMUM HEAD MOTION DURING THE IMPLIED SEEK OF THE WRITE COMMAND. THE SAME OPERATION, INCLUDING THE EXPLICIT SEEK IS REPEATED FOR READ DATA.

TEST 12 READ, WRITE CHECK W/ HEAD SWITCHING TEST

THE TEST SEEKS TO CYLINDER 559. TO INSURE THERE IS NO HEAD MOTION DURING DATA TRANSFER. THE READ DATA COMMAND STARTS WITH CYLINDER 559., TRACK 2, SECTOR 30. THE WORD COUNT IS EQUAL TO MULTIPLE SECTORS WHICH CAUSES THE SUBSYSTEM TO SWITCH FROM TRACK 2 TO TRACK 3 AFTER THE FIRST SECTOR IS WRITTEN. THE SAME SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND, USING THE SAME WORD COUNT AND STARTING SECTOR. IF THE 'SSE' BIT IS SET AFTER EXECUTING A DATA COMMAND, THE 16 BIT MODE WILL BE TERMINATED AND THE TEST IS REPEATED FOR 16 BIT FORMAT WITH THE SSEI BIT SET, WHERE SECTOR 31. IS THE LAST SECTOR. IF NO 'SSE' WAS SET, THEN BOTH FORMAT MODES WILL BE EXECUTED.

TEST 13 READ, WRITE CHECK W/ MID-TRANSFER SEEK TEST

514
515 THIS TEST READS MULTIPLE SECTORS STARTING WITH CYLINDER
516 559., LAST TRACK AND SECTOR 30., CAUSING A MID-TRANSFER SEEK
517 AFTER THE FIRST OF THE MULTIPLE SECTORS ARE WRITTEN. THE SAME
518 SECTORS ARE VERIFIED WITH A WRITE CHECK DATA COMMAND. IF THE
519 'SSE' BIT IS SET AFTER EXECUTING A DATA COMMAND, THE 16 BIT MODE
520 WILL BE TERMINATED AND THE TEST IS REPEATED FOR 16 BIT FORMAT
521 WITH THE SSEI BIT SET, WHERE SECTOR 31. IS THE LAST SECTOR. IF NO
522 'SSE' WAS SET, THEN BOTH FORMAT MODES WILL BE EXECUTED.
523
524
525
526
527
528 TEST 14 WRITE, READ W/ HCE ERROR TEST
529
530 A SECTOR IS FORMATTED WITH AN INCORRECT HEADER, THEN THE
531 TEST WRITES AND READS DATA FROM THE SECTOR AND VERIFIES THAT THE
532 CORRECT ERROR IS DETECTED. EACH BIT POSITION OF BOTH HEADER
533 WORDS IS TESTED IN THIS MANNER.
534
535
536
537
538
539 TEST 15 WRITE, READ W/ HCI & SSEI TEST
540
541 A SECTOR IS FORMATTED WITH AN INCORRECT HEADER, THEN WRITTEN
542 AND READ WITH HEADER COMPARE AND SKIP SECTOR ERROR INHIBITED.
543 THE TEST VERIFIES THAT NO ERROR IS DETECTED.
544
545
546
547
548
549 TEST 16 WRITE, READ W/ IVC ERROR TEST
550
551 VOLUME VALID IS RESET BY SETTING AND RESETTING DIAGNOSTIC
552 MODE. THE TEST THEN EXECUTES A WRITE DATA COMMAND AND VERIFIES
553 THAT INVALID COMMAND STATUS SETS. THE TEST IS R' ATED FOR READ
554 DATA COMMAND.
555
556
557
558
559
560 TEST 17 WRITE, READ W/ ABORT TEST
561
562 THE TEST SETS AN ERROR IN THE ERROR REGISTER AND EXECUTES A
563 WRITE DATA COMMAND VERIFYING THAT "PIP" REMAINS INACTIVE. THE
564 SAME PROCEDURE IS USED FOR READ DATA COMMAND.
565
566
567
568
569
570 TEST 20 WRITE, READ EARLY PEAK SHIFT TEST

```
571
572                          THIS TEST WRITES AND READS A  SINGLE  SECTOR  USING  A  DATA
573                     PATTERN DESIGNED TO INTRODUCE EARLY PEAK SHIFT.
574
575
576
577
578
579                TEST 21 WRITE, READ EACH TRACK W/ MIXED PEAK SHIFT TEST
580
581                          THE TEST WRITES AND READS WITH EACH  TRACK USING  A  MIX  OF
582                     EARLY, NORMAL AND LATE PEAK SHIFTS.
583
584
585
586
587
588                TEST 22 READ MULTIPLE SECTORS IN OFFSET MODE TEST
589
590                          THE TEST EXECUTES READ DATA AND WRITE CHECK DATA COMMANDS IN
591                     OFFSET  MODE  IN BOTH OFFSET DIRECTIONS WITH THE TRANSFER SIZE OF
592                     TWO SECTORS AT A TIME.
593
594
595
596
597
598                TEST 23 FORMAT FE CYLINDERS
599
600                          THIS  TEST  FORMATS  BOTh FE CYLINDERS  TO  RE-E.TABLISH THE
601                     CORRECT FORMAT ON THE SECTORS  THAT WERE WRITTEN UPON BY PREVIOUS
602                     TESTS. THE TEST IS DONE IN 16 BI1 FORMAT WITH THE SSEI BIT SET TO
603                     ALLOW  ALL 32 SECTORS ON EACH TRACK TO BE FORMATTED.
604
605
606
607
608
609
```

```
                        ;*LAST REVISION 08-OCT-81
                        ;
                        .TITLE  CZRNFA0 RM80 FCTNL PT3
                        ;*COPYRIGHT (C) 1982
                        ;*DIGITAL EQUIPMENT CORPORATION
                        ;*COLORADO SPGS., CO. 80919
                        ;*
                        ;*PROGRAM BY MIKE LEAVITT
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C5), 18-MAR-81
                        ;*
                        .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>
                        ;*         7            TN128
                        ;*         6            TN64
                        ;*         5            TN32
                        ;*         4            TN16
                        ;*         3            TN8
                        ;*         2            TN4
                        ;*         1            TN2
                        ;*         0            TN1
                        ;
                        .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
```

BASIC DEFINITIONS

```
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
                00001C          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
    493
    494                         .SBTTL  RM80 REGISTER BIT DEFINITIONS
    495
    496                         ;*RMCS1 CONTROL STATUS REGISTER
    497
    498         004000          DVA     = BIT11                 ;DEVICE AVAILABLE-READ ONLY
    499         000040          F4      = BIT05                 ;FUNCTION CODE
    500         000020          F3      = BIT04                 ;FUNCTION CODE
    501         000010          F2      = BIT03                 ;FUNCTION CODE
    502         000004          F1      = BIT02                 ;FUNCTION CODE
    503         000002          F0      = BIT01                 ;FUNCTION CODE
    504         000001          GO      = BIT00                 ;GO BIT
    505         000077          FNCMSK  = 000077                ;FUNCTION CODE MASK
    506
    507                         ;FUNCTION CODES (BITS 01-05 OF RMCS1)
    508         000000          NOP     = 000000                ;NOP COMMAND
    509         000002          ILF02   = 000002                ;ILLEGAL COMMAND
    510         000004          SEEK    = 000004                ;SEEK COMMAND
    511         000006          RECAL   = 000006                ;RECALIBRATE COMMAND
    512         000010          DRVCLR  = 000010                ;DRIVE CLEAR COMMAND
    513         000012          RLEASE  = 000012                ;RELEASE COMMAND
    514         000014          OFFSET  = 000014                ;OFFSET COMMAND
    515         000016          RTC     = 000016                ;RETURN TO CENTERLINE COMMAND
    516         000020          RIP     = 000020                ;READ IN PRESET COMMAND
    517         000022          PAKACK  = 000022                ;PACK ACKNOWLEDGE COMMAND
    518         000022          PACACK  = PAKACK
    519         000024          ILF24   = 000024                ;ILLEGAL COMMAND
    520         000026          ILF26   = 000026                ;ILLEGAL COMMAND
```

RM80 REGISTER BIT DFFINITIONS

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

                                ;*RMDA  DISK ADDRESS REGISTER

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

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

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

                                ;*RMDS  DRIVE STATUS REGISTER

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

                                ;*RMER1 ERROR REGISTER #1
```

RM80 REGISTER BIT DEFINITIONS

```
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" EPROR
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
629     040000          RG      = BIT14                 ;RUN AND GO
```

RM80 REGISTER BIT DEFINITIONS

```
630          020000                  EBL     = BIT13                 ;END OF BLOCK
631          010000                  REX     = BIT12                 ;EXCEPTION
632          004000                  ESRC    = BIT11                 ;ENABLE SEARCH
633          002000                  PLFS    = BIT10                 ;LOOKING FOR SYNC
634          001000                  ECRC    = BIT09                 ;ENABLE CRC OUT
635          000400                  PDA     = BIT08                 ;DATA AREA
636          000200                  PHA     = BIT07                 ;HEADER AREA
637          000100                  CONT    = BIT06                 ;CONTINUE
638          000040                  WC      = BIT05                 ;WORD CLOCK
639          000020                  EECC    = BIT04                 ;ENABLE ECC OUT
640          000010                  MWD     = BIT03                 ;WRITE DATA BIT
641          000004                  LS      = BIT02                 ;LAST SECTOR
642          000002                  LST     = BIT01                 ;LAST SECTOR AND TRACK
643          000001                  DMD     = BIT00                 ;DIAGNOSTIC MODE
644
645                                  ;*RMDT  DRIVE TYPE REGISTER
646
647          100000                  NSA     = BIT15                 ;NOT SECTOR ADDRESSED = 0
648          040000                  TAP     = BIT14                 ;TAPE DRIVE = 0
649          020000                  MOH     = BIT13                 ;MOVING HEAD = 1
650          004000                  DRQ     = BIT11                 ;DRIVE REQUEST REQUIRED
651
652          020026                  SNGPRT  = 020026                ;SINGLE PORT DRIVE TYPE
653          024026                  DULPRT  = 024026                ;DUAL PORT DRIVE TYPE
654
655                                  ;*RMOF  OFFSET REGISTER
656
657          010000                  FMT16   = BIT12                 ;16 BIT WORD FORMAT
658          004000                  ECI     = BIT11                 ;ECC INHIBIT
659          002000                  HCI     = BIT10                 ;HEADER COMPARE INHIBIT
660          001000                  SSEI    = BIT09                 ;SKIP SECTOR ERROR INHIBIT
661          000200                  OFD     = BIT07                 ;OFFSET FORWARD
662
663                                  ;*RMDC  DESIRED CYLINDER ADDRESS REGISTER
664
665          001777                  CYLMSK  = 001777                ;MASK FOR CYLINDER ADDRESS
666
667                                  ;*RMMR2 MAINTENANCE REGISTER #2
668
669                                  ;READ ONLY BITS
670          100000                  RQA     = BIT15                 ;PORT A REQUEST
671          040000                  RQB     = BIT14                 ;PORT B REQUEST
672          020000                  TAG     = BIT13                 ;TAG CONTROL
673          010000                  TST     = BIT12                 ;COMMAND SEQUENCE TEST BIT
674          004000                  CC      = BIT11                 ;CONTROL OR CYLINDER TAG
675          002000                  CH      = BIT10                 ;CONTROL OR HEAD TAG
678          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
679
```

```
                                ;*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
000040                          SSE     = BIT05                 ;SKIP SECTOR ERROR
000010                          DPE     = BIT03                 ;DATA PARITY ERROR

                                .SBTTL  PROGRAM MNEMONICS

100000                          MSE     = BIT15                 ;MANUFACTURING DETECTED SECTOR ERROR
040000                          USE     = BIT14                 ;USER DETECTED SECTOR ERROR
020000                          SSF     = BIT13                 ;SKIP SECTOR FAILURE

                                .SBTTL  RM80 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
```

```
RH CONTROLLER REGISTER BIT DEFINITIONS

                        MCPE    = BIT13                 ;MASSBUS CONTROL BUS PARITY ERROR-READ ONLY
                        PSEL    = BIT10                 ;PORT B SELECT
                        A17     = BIT09                 ;ADDRESS EXTENSION
                        A16     = BIT08                 ;ADDRESS EXTENSION
                        RDY     = BIT07                 ;READY-READ ONLY
                        IE      = BIT06                 ;INTERRUPT ENABLE

                        ;*RMCS2 RH CONTROL STATUS REGISTER #2

                        DLT     = BIT15                 ;DATA LATE-READ ONLY
                        WCE     = BIT14                 ;WRITE CHECK ERROR-READ ONLY
                        UPE     = BIT13                 ;UNIBUS PARITY ERROR
                        NED     = BIT12                 ;NONEXISTANT DRIVE-READ ONLY
                        NEM     = BIT11                 ;NONEXISTANT MEMORY-READ ONLY
                        PGE     = BIT10                 ;PROGRAM ERROR-READ ONLY
                        MXF     = BIT09                 ;MISSED TRANSFER
                        MDPE    = BIT08                 ;MASSBUS DATA BUS PARITY ERROR-READ ONLY
                        OR      = BIT07                 ;OUTPUT READY-READ ONLY
                        IR      = BIT06                 ;INPUT READY-READ ONLY
                        CLR     = BIT05                 ;CONTROLLER CLEAR
                        PAT     = BIT04                 ;PARITY TEST
                        BAI     = BIT03                 ;UNIBUS ADDRESS INCREMENT INHIBIT
                        U2      = BIT02                 ;UNIT SELECT
                        U1      = BIT01                 ;UNIT SELECT
                        U0      = BIT00                 ;UNIT SELECT

                        ;UNIT SELECT MASK

                        UNTMSK  = 7                     ;UNIT SELECT MASK

                        ;*RMCS3 RH70 CONTROL STATUS REGISTER #3

                        APE     = BIT15                 ;ADDRESS PARITY ERROR
                        DPEHI   = BIT14                 ;DATA PARITY ERROR HIGH WORD
                        DPELO   = BIT13                 ;DATA PARITY ERROR LOW WORD
                        WCEHI   = BIT12                 ;WRITE CHECK ERROR HIGH WORD
                        WCELO   = BIT11                 ;WRITE CHECK ERROR LOW WORD
                        DBL     = BIT10                 ;DOUBLE WORD TRANSFER
                        IE      = BIT06                 ;INTERRUPT ENABLE
                        IPCK3   = BIT03                 ;INVERT PARITY CHECK
                        IPCK2   = BIT02                 ;INVERT PARITY CHECK
                        IPCK1   = BIT01                 ;INVERT PARITY CHECK
                        IPCK0   = BIT00                 ;INVERT PARITY CHECK

                        .SBTTL  RH CONTROLLER REGISTER INDEX VALUES

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

                        ABASE   = 176700                ;UNIBUS ADDRESS
                        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  005434               JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
2
3 000204  000137  005424               JMP     @#START1        ;CHANGE RH/RM BUS ADDRESS
4
5                               .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000210          $SVPC=.                 ;SAVE PC
                        000046          .=46
  000046  033756                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
  000052  000000                        .WORD   0               ;;2)SET LOC.52 TO ZERO
                        000210          .=$SVPC                 ;; RESTORE PC
6
7               001100          .=1100
8                               .SBTTL  APT PARAMETER BLOCK

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

  001100                        $APTHD:
  001100  000000        $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
  001102  001222        $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
  001104  000024        $TSTM:  .WORD   20.     ;;RUN TIM OF LONGEST TEST
  001106  000024        $PASTM: .WORD   20.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
  001110  000024        $UNITM: .WORD   20.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDED UNIT
  001112  000042                .WORD   $ETEND-$MAIL/2  ;;LENGTH MAILBOX-ETABLE(WORDS)
9               001114          TAGADR=.
```

```
      0                                 .SBTTL  COMMON TAGS

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

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

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

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

```
                                .SBTTL  USER DEFINED TAGS

001326  000000                  CTLFG:  .WORD   0       ;CONTAINS CONTROL-C FLAG
001330  000000                  CHGADR: .WORD   0       ;CHANGE RH/RM BUS ADDRESS = -1, NO CHANGE = 0
001332  000000                  XXDP:   .WORD   0       ;THE LOW BYTE CONTAINS THE DRIVE NUMBER FROM WHICH
                                                        ;THE PROGRAM WAS LOADED. THE HIGH BYTE CONTAINS THE
                                                        ;'XXDP' DEVICE CODE FOR THE RM80.
001334      000                 LSTRK:  .BYTE   0       ;LO BYTE = 0
001335      015                         .BYTE   13.     ;HI BYTE, CONTAINS LAST TRACK ADDRESS OF UNIT
                                                        ;UNDER TEST. RM80 = 13.

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

001336                          GETBUF:

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

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

001412                          PUTBUF:

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

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

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

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

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

                                ;SKIP SECTOR FAILURE ENABLE IS SET IF THE "SSF" BIT SHOULD BE
                                ;SET IN THE FIRST HEADER WORD WHEN THE FE CYLINDERS ARE BEING
                                ;REFORMATTED.
001514  000000                  SSFENB: .WORD   0               ;SSF ENABLE

                                ;LOCATIONS "ASNDC" AND "ASNDC" CONTAIN THE CYLINDER, TRACK AND SECTOR
                                ;ADDRESS ASSIGNED BY THE BAD SECTOR MODULE.
001516  000000                  ASNDC:  .WORD   0               ;ASSIGNED DESIRED CYLINDER
001520  000000                  ASNDA:  .WORD   0               ;ASSIGNED TRACK, AND SECTOR
001522  000000                  CLKADR: .WORD   0               ;UNIBUS ADDRESS OF KW11 CLOCK
001524  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.
001526                          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.
001555                          PUTINX: .BLKB   23.             ;PUT INDEX TABLE

                                ;PUT TAGS HERE
```

ERROR POINTER TABLE

```
                                .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


  001604                        $ERRTB:
1
2                               ;ERROR  1       WRONG UNIT SELECTED
3
  001604  070002                        EMT1
  001606  074140                        EHT1
  001610  074264                        EDT1
  001612  074354                        EFT1

4
5                               ;ERROR  2       DEVICE WENT UNAVAILABLE
6
  001614  070006                        EMT2
  001616  074140                        EHT1
  001620  074264                        EDT1
  001622  074354                        EFT1

7
8                               ;ERROR  3       DEVICE WENT NONEXISTENT
9
  001624  070014                        EMT3
  001626  074140                        EHT1
  001630  074264                        EDT1
  001632  074354                        EFT1

10
11                              ;ERROR  4       CONTROLLER NOT READY
12
  001634  070022                        EMT4
  001636  074140                        EHT1
  001640  074264                        EDT1
  001642  074354                        EFT1

13
14                              ;ERROR  5       DRIVE NOT READY AND GO NOT RESET
15
  001644  070030                        EMT5
  001646  074140                        EHT1
  001650  074264                        EDT1
  001652  074354                        EFT1

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

ERROR POINTER TABLE

```
     18
        001654  070036                          EMT6
        001656  074140                          EHT1
        001660  074264                          EDT1
        001662  074354                          EFT1

     19
     20                             ;ERROR  7       BUS TIMEOUT TRYING TO READ OR WRITE REGISTER
     21
        001664  070044                          EMT7
        001666  000000                          0
        001670  000000                          0
        001672  000000                          0

     22
     23                             ;ERROR  10      DRIVE NOT READY BUT GO IS RESET
     24
        001674  070052                          EMT10
        001676  074140                          EHT1
        001700  074264                          EDT1
        001702  074354                          EFT1

     25
     26                             ;ERROR  11      GO NOT RESET BUT DRIVE IS READY
     27
        001704  070056                          EMT11
        001706  074140                          EHT1
        001710  074264                          EDT1
        001712  074354                          EFT1

     28
     29                             ;ERROR  12      INCORRECT FUNCTION CODE
     30
        001714  070062                          EMT12
        001716  074140                          EHT1
        001720  074264                          EDT1
        001722  074354                          EFT1

     31
     32                             ;ERROR  13      PARITY ERROR READING REMOTE REGISTERS
     33
        001724  070070                          EMT13
        001726  074140                          EHT1
        001730  074264                          EDT1
        001732  074354                          EFT1

     34
     35                             ;ERROR  14      TRANSFER ERROR IS INCORRECT
     36
        001734  070102                          EMT14
        001736  074140                          EHT1
        001740  074264                          EDT1
        001742  074354                          EFT1

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

ERROR POINTER TABLE

```
        001744  070110                          EMT15
        001746  074140                          EHT1
        001750  074264                          EDT1
        001752  074354                          EFT1

    40
    41                          ;ERROR  16      INCORRECT BUS ADDRESS
    42
        001754  070116                          EMT16
        001756  074140                          EHT1
        001760  074264                          EDT1
        001762  074354                          EFT1

    43
    44                          ;ERROR  17      INCORRECT LBT STATUS
    45
        001764  070126                          EMT17
        001766  074140                          EHT1
        001770  074264                          EDT1
        001772  074354                          EFT1

    46
    47                          ;ERROR  20      INCORRECT AOE
    48
        001774  070136                          EMT20
        001776  074140                          EHT1
        002000  074264                          EDT1
        002002  074354                          EFT1

    49
    50                          ;ERROR  21      INCORRECT DISK ADDRESS
    51
        002004  070146                          EMT21
        002006  074140                          EHT1
        002010  074264                          EDT1
        002012  074354                          EFT1

    52
    53                          ;ERROR  22      INCORRECT CYLINDER ADDRESS
    54
        002014  070156                          EMT22
        002016  074140                          EHT1
        002020  074264                          EDT1
        002022  074354                          EFT1

    55
    56                          ;ERROR  23      INCORRECT WLE STATUS
    57
        002024  070166                          EMT23
        002026  074140                          EHT1
        002030  074264                          EDT1
        002032  074354                          EFT1

    58
    59                          ;ERROR  24      INCORRECT UPE STATUS
    60
        002034  070176                          EMT24
```

ERROR POINTER TABLE

```
002036  074140                          EHT1
002040  074264                          EDT1
002042  074354                          EFT1

61
62                              ;ERROR  25      INCORRECT WCF STATUS
63
002044  070206                          EMT25
002046  074140                          EHT1
002050  074264                          EDT1
002052  074354                          EFT1

64
65                              ;ERROR  26      INCORRECT WCE STATUS
66
002054  070216                          EMT26
002056  074140                          EHT1
002060  074264                          EDT1
002062  074354                          EFT1

67
68                              ;ERROR  27      INCORRECT MDPE STATUS
69
002064  070226                          EMT27
002066  074140                          EHT1
002070  074264                          EDT1
002072  074354                          EFT1

70
71                              ;ERROR  30      INCORRECT DCK STATUS
72
002074  070236                          EMT30
002076  074140                          EHT1
002100  074264                          EDT1
002102  074354                          EFT1

73
74                              ;ERROR  31      INCORRECT ECH STATUS
75
002104  070246                          EMT31
002106  074140                          EHT1
002110  074264                          EDT1
002112  074354                          EFT1

76
77                              ;ERROR  32      DLT SHOULD NOT BE SET
78
002114  070256                          EMT32
002116  074140                          EHT1
002120  074264                          EDT1
002122  074354                          EFT1

79
80                              ;ERROR  33      MXF SHOULD NOT BE SET
81
002124  070266                          EMT33
002126  074140                          EHT1
```

```
002130  074264                          EDT1
002132  074354                          EFT1

                                ;ERROR  34      DTE SHOULD NOT BE SET

002134  070276                          EMT34
002136  074140                          EHT1
002140  074264                          EDT1
002142  074354                          EFT1

                                ;ERROR  35      INCORRECT HCRC STATUS

002144  070306                          EMT35
002146  074140                          EHT1
002150  074264                          EDT1
002152  074354                          EFT1

                                ;ERROR  36      INCORRECT HCE STATUS

002154  070316                          EMT36
002156  074140                          EHT1
002160  074264                          EDT1
002162  074354                          EFT1

                                ;ERROR  37      INCORRECT FER STATUS

002164  070326                          EMT37
002166  074140                          EHT1
002170  074264                          EDT1
002172  074354                          EFT1

                                ;ERROR  40      DPE SHOULD NOT BE SET (NOT A DATA COMMAND)

002174  070336                          EMT40
002176  074140                          EHT1
002200  074264                          EDT1
002202  074354                          EFT1

                                ;ERROR  41      LOST 'MOL' DURING PACK ACKNOWLEDGE

002204  070344                          EMT41
002206  074140                          EHT1
002210  074264                          EDT1
002212  074354                          EFT1

                                ;ERROR  42      UNSAFE ERROR DURING PACK ACKNOWLEDGE

002214  070354                          EMT42
002216  074140                          EHT1
002220  074264                          EDT1
```

ERROR POINTER TABLE

```
                002222  074354                          EFT1

    103
    104                                 ;ERROR  43      "DPI" ERROR DURING PACK ACKNOWLEDGE
    105
                002224  070366                          EMT43
                002226  074140                          EHT1
                002230  074264                          EDT1
                002232  074354                          EFT1

    106
    107                                 ;ERROR  44      "RMR" ERROR DURING PACK ACKNOWLEDGE
    108
                002234  070376                          EMT44
                002236  074140                          EHT1
                002240  074264                          EDT1
                002242  074354                          EFT1

    109
    110                                 ;ERROR  45      "ILR" ERROR DURING PACK ACKNOWLEDGE
    111
                002244  070406                          EMT45
                002246  074140                          EHT1
                002250  074264                          EDT1
                002252  074354                          EFT1

    112
    113                                 ;ERROR  46      "ILF" ERROR DURING PACK ACKNOWLEDGE
    114
                002254  070416                          EMT46
                002256  074140                          EHT1
                002260  074264                          EDT1
                002262  074354                          EFT1

    115
    116                                 ;ERROR  47      COMPOSITE ERROR STATUS IS INCORRECT
    117
                002264  070426                          EMT47
                002266  074140                          EHT1
                002270  074264                          EDT1
                002272  074354                          EFT1

    118
    119                                 ;ERROR  50      PARITY ERROR WRITING REMOTE REGISTERS
    120
                002274  070434                          EMT50
                002276  074140                          EHT1
                002300  074264                          EDT1
                002302  074354                          EFT1

    121
    122                                 ;ERROR  51      INCORRECT IAE STATUS DURING SEEK COMMAND
    123
                002304  070444                          EMT51
                002306  074140                          EHT1
                002310  074264                          EDT1
                002312  074354                          EFT1
```

ERROR POINTER TABLE

```
124
125                                     ;ERROR  52      OPI ERROR DURING SEEK - MEDIUM IS NOT ON LINE
126
     002314  070456                             EMT52
     002316  074140                             EHT1
     002320  074264                             EDT1
     002322  074354                             EFT1

127
128                                     ;ERROR  53      OPI ERROR DURING SEEK - MEDIUM IS ON LINE, ASSUME
129                                     ;               ON CYLINDER LATCH DIDN'T RESET
130
     002324  070474                             EMT53
     002326  074140                             EHT1
     002330  074264                             EDT1
     002332  074354                             EFT1

131
132                                     ;ERROR  54      SEEK INCOMPLETE ERROR DURING SEEK COMMAND
133
     002334  070512                             EMT54
     002336  074140                             EHT1
     002340  074264                             EDT1
     002342  074354                             EFT1

134
135                                     ;ERROR  55      DEVICE CHECK DURING SEEK COMMAND
136
     002344  070522                             EMT55
     002346  074140                             EHT1
     002350  074264                             EDT1
     002352  074354                             EFT1

137
138                                     ;ERROR  56      PIP IS STILL SET AFTER SEEK - SKI IS RESET
139
     002354  070534                             EMT56
     002356  074140                             EHT1
     002360  074264                             EDT1
     002362  074354                             EFT1

140
141                                     ;ERROR  57      ATA DID NOT SET DURING SEEK COMMAND
142
     002364  070552                             EMT57
     002366  074140                             EHT1
     002370  074264                             EDT1
     002372  074354                             EFT1

143
144                                     ;ERROR  60      IVC ERROR DURING SEEK COMMAND - LOST
145                                     ;               VOLUME VALID
146
     002374  070562                             EMT60
     002376  074140                             EHT1
     002400  074264                             EDT1
```

ERROR POINTER TABLE

```
        002402  074354                          EFT1

    147
    148                             ;ERROR  61      ERRONEOUS IVC ERROR DURING SEEK COMMAND -
    149                             ;               VOLUME VALID IS STIL SET
    150
        002404  070600                          EMT61
        002406  074140                          EHT1
        002410  074264                          EDT1
        002412  074354                          EFT1

    151
    152                             ;ERROR  62      MOL IS ZERO, BUT OPI WAS NOT
    153                             ;               REPORTED DURING SEEK COMMAND
    154
        002414  070620                          EMT62
        002416  074140                          EHT1
        002420  074264                          EDT1
        002422  074354                          EFT1

    155
    156                             ;ERROR  63      SSE SHOULD NOT BE SET
    157
        002424  070634                          EMT63
        002426  074140                          EHT1
        002430  074264                          EDT1
        002432  074354                          EFT1

    158
    159                             ;ERROR  64      DRIVE DID NOT DETECT "IVC" ERROR DURING SEEK
    160
        002434  070644                          EMT64
        002436  074140                          EHT1
        002440  074264                          EDT1
        002442  074354                          EFT1

    161
    162                             ;ERROR  65      DRIVE EXECUTED A SEEK WITH ERROR SET
    163
        002444  070664                          EMT65
        002446  074140                          EHT1
        002450  074264                          EDT1
        002452  074354                          FFT1

    164
    165                             ;ERROR  66      UNEXPECTED ERROR SET IN RMER1
    166
        002454  070704                          EMT66
        002456  074140                          EHT1
        002460  074264                          EDT1
        002462  074354                          EFT1

    167
    168                             ;ERROR  67      UNEXPECTED ERROR SET IN RMER2
    169
        002464  070716                          EMT67
        002466  074140                          EHT1
```

ERROR POINTER TABLE

```
002470  074264                          EDT1
002472  074354                          EFT1

170
171                             ;ERROR  70      ERRONEOUS ''IAE'' ERROR DURING RECALIBRATE
172
002474  070730                          EMT70
002476  074140                          EHT1
002500  074264                          EDT1
002502  074354                          EFT1

173
174                             ;ERROR  71      ''ILF'' ERROR DURING RECALIBRATE
175
002504  070740                          EMT71
002506  074140                          EHT1
002510  074264                          EDT1
002512  074354                          EFT1

176
177                             ;ERROR  72      ''OPI'' ERROR DURING RECALIBRATE DUE TO ''MOL'' = 0
178
002514  070750                          EMT72
002516  074140                          EHT1
002520  074264                          EDT1
002522  074354                          EFT1

179
180                             ;ERROR  73      ''OPI'' ERROR DURING RECALIBRATE BECAUSE ON
181                             ;               CYLINDER DIDNT DROP
182
002524  070766                          EMT73
002526  074140                          EHT1
002530  074264                          EDT1
002532  074354                          EFT1

183
184                             ;ERROR  74      ''IVC'' ERROR DURING RECALIBRATE - ''VV'' = 0
185
002534  071004                          EMT74
002536  074140                          EHT1
002540  074264                          EDT1
002542  074354                          EFT1

186
187                             ;ERROR  75      ERRONEOUS ''IVC'' ERROR DURING RECALIBRATE - ''VV'' = 1
188
002544  071014                          EMT75
002546  074140                          EHT1
002550  074264                          EDT1
002552  074354                          EFT1

189
190                             ;ERROR  76      ''SKI'' ERROR DURING RECALIBRATE
191
002554  071034                          EMT76
002556  074140                          EHT1
```

ERROR POINTER TABLE

```
        002560  074264                          EDT1
        002562  074354                          EFT1

    192
    193                             ;ERROR  77      'DVC'' OCCURRED DURING RECALIBRATE
    194
        002564  071044                          EMT77
        002566  074140                          EHT1
        002570  074264                          EDT1
        002572  074354                          EFT1

    195
    196                             ;ERROR  100     LOST 'MOL'' DURING RECALIBRATE - ''OPI'' = 0
    197
        002574  071056                          EMT100
        002576  074140                          EHT1
        002600  074264                          EDT1
        002602  074354                          EFT1

    198
    199                             ;ERROR  101     LOST 'VV'' DURING RECALIBRATE - ''IVC'' = 0
    200
        002604  071074                          EMT101
        002606  074140                          EHT1
        002610  074264                          EDT1
        002612  074354                          EFT1

    201
    202                             ;ERROR  102     ''ATA'' DID NOT SET DURING RECALIBRATE
    203
        002614  071112                          EMT102
        002616  074140                          EHT1
        002620  074264                          EDT1
        002622  074354                          EFT1

    204
    205                             ;ERROR  103     'OM'' DID NOT RESET DURING RECALIBRATE
    206
        002624  071122                          EMT103
        002626  074140                          EHT1
        002630  074264                          EDT1
        002632  074354                          EFT1

    207
    208                             ;ERROR  104     'PIP'' IS STIL SET AFTER RECALIBRATE
    209
        002634  071134                          EMT104
        002636  074140                          EHT1
        002640  074264                          EDT1
        002642  074354                          EFT1

    210
    211                             ;ERROR  105     UNEXPECTED ''ILR'' ERROR DURING RECALIBRATE
    212
        002644  071152                          EMT105
        002646  074140                          EHT1
        002650  074264                          EDT1
```

ERROR POINTER TABLE

```
002652  074354                          EFT1

213
214                             ;ERROR  106     UNEXPECTED ''RMR'' ERROR DURING RECALIBRATE
215
002654  071162                          EMT106
002656  074140                          EHT1
002660  074264                          EDT1
002662  074354                          EFT1

216
217                             ;ERROR  107     ''UNS'' ERROR DURING RECALIBRATE - AC POWER IS LOW
218
002664  071172                          EMT107
002666  074140                          EHT1
002670  074264                          EDT1
002672  074354                          EFT1

219
220                             ;ERROR  110     CANNOT ACCESS MASSBUS CONTROLLER VIA UNIBUS
221
002674  071212                          EMT110
002676  074162                          EHT110
002700  074302                          EDT110
002702  074372                          EFT110

222
223                             ;ERROR  111     NONEXISTENT DEVICE
224
002704  071224                          EMT111
002706  074166                          EHT111
002710  074304                          EDT111
002712  074374                          EFT111

225
226                             ;ERROR  112     DEVICE NOT AVAILABLE
227
002714  071232                          EMT112
002716  074166                          EHT111
002720  074304                          EDT111
002722  074374                          EFT111

228
229                             ;ERROR  113     BUS TIMEOUT-NED STATUS FAILURE
230
002724  071240                          EMT113
002726  000000                          0
002730  000000                          0
002732  000000                          0

231
232                             ;ERROR  114     UNUSED
233
002734  000000                          0
002736  000000                          0
002740  000000                          0
002742  000000                          0
```

ERROR POINTER TABLE

```
234
235                                     ;ERROR  115     RMCS1 NOT INITIALIZED BY UNIBUS
236
     002744  071256                             EMT115
     002746  074140                             EHT1
     002750  074264                             EDT1
     002752  074354                             EFT1

237
238                                     ;ERROR  116     RMBA NOT INITIALIZED BY UNIBUS
239
     002754  071266                             EMT116
     002756  074140                             EHT1
     002760  074264                             EDT1
     002762  074354                             EFT1

240
241                                     ;ERROR  117     RMCS2 NOT INITIALIZED BY UNIBUS
242
     002764  071276                             EMT117
     002766  074140                             EHT1
     002770  074264                             EDT1
     002772  074354                             EFT1

243
244                                     ;ERROR  120     RMER1 NOT INITIALIZED BY UNIBUS
245
     002774  071306                             EMT120
     002776  074140                             EHT1
     003000  074264                             EDT1
     003002  074354                             EFT1

246
247                                     ;ERROR  121     RMAS NOT INITIALIZED BY UNIBUS
248
     003004  071316                             EMT121
     003006  074140                             EHT1
     003010  074264                             EDT1
     003012  074354                             EFT1

249
250                                     ;ERROR  122     RMMR1 NOT INITIALIZED BY UNIBUS
251
     003014  071326                             EMT122
     003016  074140                             EHT1
     003020  074264                             EDT1
     003022  074354                             EFT1

252
253                                     ;ERROR  123     RMDS NOT INITIALIZED BY UNIBUS
254
     003024  071336                             EMT123
     003026  074140                             EHT1
     003030  074264                             EDT1
     003032  074354                             EFT1
```

ERROR POINTER TABLE

```
255
256                                     ;ERROR  124     RMEC2 NOT INITIALIZED BY UNIBUS
257
    003034  071346                             EMT124
    003036  074140                             EHT1
    003040  074264                             EDT1
    003042  074354                             EFT1

258
259                                     ;ERROR  125     RMMR2 NOT INITIALIZED BY UNIBUS
260
    003044  071356                             EMT125
    003046  074140                             EHT1
    003050  074264                             EDT1
    003052  074354                             EFT1

261
262                                     ;ERROR  126     RMCS1 NOT CLEARED BY CONTROLLER CLEAR
263
    003054  071366                             EMT126
    003056  074140                             EHT1
    003060  074264                             EDT1
    003062  074354                             EFT1

264
265                                     ;ERROR  127     RMBA NOT CLEARED BY CONTROLLER CLEAR
266
    003064  071400                             EMT127
    003066  074140                             EHT1
    003070  074264                             EDT1
    003072  074354                             EFT1

267
268                                     ;ERROR  130     RMCS2 NOT CLEARED BY CONTROLLER CLEAR
269
    003074  071412                             EMT130
    003076  074140                             EHT1
    003100  074264                             EDT1
    003102  074354                             EFT1

270
271                                     ;ERROR  131     RMER1 NOT CLEARED BY CONTROLLER CLEAR
272
    003104  071424                             EMT131
    003106  074140                             EHT1
    003110  074264                             EDT1
    003112  074354                             EFT1

273
274                                     ;ERROR  132     RMAS NOT CLEARED BY CONTROLLER CLEAR
275
    003114  071436                             EMT132
    003116  074140                             EHT1
    003120  074264                             EDT1
    003122  074354                             EFT1

276
```

ERROR POINTER TABLE

```
                                        ;ERROR  133     RMMR1 NOT CLEARED BY CONTROLLER CLEAR
003124  071450                                  EMT133
003126  074140                                  EHT1
003130  074264                                  EDT1
003132  074354                                  EFT1

                                        ;ERROR  134     RMDS NOT CLEARED BY CONTROLLER CLEAR
003134  071462                                  EMT134
003136  074140                                  EHT1
003140  074264                                  EDT1
003142  074354                                  EFT1

                                        ;ERROR  135     RMEC2 NOT CLEARED BY CONTROLLER CLEAR
003144  071474                                  EMT135
003146  074140                                  EHT1
003150  074264                                  EDT1
003152  074354                                  EFT1

                                        ;ERROR  136     RMMR2 NOT CLEARED BY CONTROLLER CLEAR
003154  071506                                  EMT136
003156  074140                                  EHT1
003160  074264                                  EDT1
003162  074354                                  EFT1

                                        ;ERROR  137     RMCS1 NOT CLEARED BY ERROR CLEAR
003164  071520                                  EMT137
003166  074140                                  EHT1
003170  074264                                  EDT1
003172  074354                                  EFT1

                                        ;ERROR  140     RMCS2 NOT CLEARED BY ERROR CLEAR
003174  071530                                  EMT140
003176  074140                                  EHT1
003200  074264                                  EDT1
003202  074354                                  EFT1

                                        ;ERROR  141     RMCS1 NOT CLEARED BY DRIVE CLEAR
003204  071540                                  EMT141
003206  074140                                  EHT1
003210  074264                                  EDT1
003212  074354                                  EFT1

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

ERROR POINTER TABLE

```
299
     003214  071550                          EMT142
     003216  074140                          EHT1
     003220  074264                          EDT1
     003222  074354                          EFT1

300
301                                 ;ERROR  143     RMER1 NOT CLEARED BY DRIVE CLEAR
302
     003224  071560                          EMT143
     003226  074140                          EHT1
     003230  074264                          EDT1
     003232  074354                          EFT1

303
304                                 ;ERROR  144     RMAS NOT CLEARED BY DRIVE CLEAR
305
     003234  071570                          EMT144
     003236  074140                          EHT1
     003240  074264                          EDT1
     003242  074354                          EFT1

306
307                                 ;ERROR  145     RMMR1  NOT CLEARED BY DRIVE CLEAR
308
     003244  071600                          EMT145
     003246  074140                          EHT1
     003250  074264                          EDT1
     003252  074354                          EFT1

309
310                                 ;ERROR  146     RMMR2 NOT CLEARED BY DRIVE CLEAR
311
     003254  071610                          EMT146
     003256  074140                          EHT1
     003260  074264                          EDT1
     003262  074354                          EFT1

312
313                                 ;ERROR  147     RMER2 NOT CLEARED BY DRIVE CLEAR
314
     003264  071620                          EMT147
     003266  074140                          EHT1
     003270  074264                          EDT1
     003272  074354                          EFT1

315
316                                 ;ERROR  150     RMEC2 NOT CLEARED BY DRIVE CLEAR
317
     003274  071630                          EMT150
     003276  074140                          EHT1
     003300  074264                          EDT1
     003302  074354                          EFT1

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

ERROR POINTER TABLE

```
        003304  071640                          EMT151
        003306  074140                          EHT1
        003310  074264                          EDT1
        003312  074354                          EFT1

    321
    322                             ;ERROR  152     DRIVE FAULT
    323
        003314  071652                          EMT152
        003316  074140                          EHT1
        003320  074264                          EDT1
        003322  074354                          EFT1

    324
    325                             ;ERROR  153     UNSAFE SHOULD BE SET BECAUSE DVC IS SET
    326
        003324  071664                          EMT153
        003326  074140                          EHT1
        003330  074264                          EDT1
        003332  074354                          EFT1

    327
    328                             ;ERROR  154     UNSAFE SHOULD NOT BE SET, AC IS LOW
    329
        003334  071702                          EMT154
        003336  074140                          EHT1
        003340  074264                          EDT1
        003342  074354                          EFT1

    330
    331                             ;ERROR  155     VOLUME VALID NOT SET BY PACK ACK
    332
        003344  071720                          EMT155
        003346  074140                          EHT1
        003350  074264                          EDT1
        003352  074354                          EFT1

    333
    334                             ;ERROR  156     OFFSET MODE NOT SET BY OFFSET COMMAND
    335
        003354  071732                          EMT156
        003356  074140                          EHT1
        003360  074264                          EDT1
        003362  074354                          EFT1

    336
    337                             ;ERROR  157     OFFSET MODE NOT RESET BY RTC COMMAND
    338
        003364  071744                          EMT157
        003366  074140                          EHT1
        003370  074264                          EDT1
        003372  074354                          EFT1

    339
    340                             ;ERROR  160     RMOF NOT RESET BY RIP COMMAND
    341
        003374  071756                          EMT160
```

ERROR POINTER TABLE

```
003376  074140                          EHT1
003400  074264                          EDT1
003402  074354                          EFT1

342
343                             ;ERROR  161     RMDA NOT RESET BY RIP COMMAND
344
003404  071766                          EMT161
003406  074140                          EHT1
003410  074264                          EDT1
003412  074354                          EFT1

345
346                             ;ERROR  162     RMDC NOT RESET BY RIP COMMAND
347
003414  072000                          EMT162
003416  074140                          EHT1
003420  074264                          EDT1
003422  074354                          EFT1

348
349                             ;ERROR 163      DATA WAS ECC CORRECTED BUT DOES NOT COMPARE WITH
350                             ;               WRITE BUFFER
351
003424  073734                          EMT336
003426  074222                          EHT336
003430  074320                          EDT336
003432  074410                          EFT336

352
353                             ;ERROR  164     OPI SHOULD NOT BE SET
354
003434  072022                          EMT164
003436  074140                          EHT1
003440  074264                          EDT1
003442  074354                          EFT1

355
356                             ;ERROR  165     IVC SHOULD NOT BE SET
357
003444  072030                          EMT165
003446  074140                          EHT1
003450  074264                          EDT1
003452  074354                          EFT1

358
359                             ;ERROR  166     IAE SHOULD NOT BE SET
360
003454  072036                          EMT166
003456  074140                          EHT1
003460  074264                          EDT1
003462  074354                          EFT1

361
362                             ;ERROR  167     NEM SHOULD NOT BE SET
363
003464  072044                          EMT167
```

ERROR POINTER TABLE

```
003466  074140                          EHT1
003470  074264                          EDT1
003472  074354                          EFT1

364
365                             ;ERROR  170     INCORRECT "MOL" STATUS DURING DIAGNOSTIC MODE
366
003474  072052                          EMT170
003476  074140                          EHT1
003500  074264                          EDT1
003502  074354                          EFT1

367
368                             ;ERROR  171     "ATA" NOT SET DURING RETURN TO CENTERLINE
369
003504  072064                          EMT171
003506  074140                          EHT1
003510  074264                          EDT1
003512  074354                          EFT1

370
371                             ;ERROR  172     "ATA" NOT SET BY OFFSET COMMAND
372
003514  072074                          EMT172
003516  074140                          EHT1
003520  074264                          EDT1
003522  074354                          EFT1

373
374                             ;ERROR  173     RMER2 NOT INITIALIZED BY UNIBUS INIT
375
003524  072104                          EMT173
003526  074140                          EHT1
003530  074264                          EDT1
003532  074354                          EFT1

376
377                             ;ERROR  174     RMER2 NOT INITIALIZED BY CONTROLLER CLEAR
378
003534  072114                          EMT174
003536  074140                          EHT1
003540  074264                          EDT1
003542  074354                          EFT1

379
380                             ;ERROR  175     SELECTED DEVICE IS IN WRITE PROTECT
381
003544  072126                          EMT175
003546  074140                          EHT1
003550  074264                          EDT1
003552  074354                          EFT1

382
383                             ;ERROR  176     CANNOT SET DIAGNOSTIC MODE
384
003554  072134                          EMT176
003556  074140                          EHT1
```

ERROR POINTER TABLE

```
003550  074264                          EDT1
003562  074354                          EFT1

385
386                             ;ERROR  177     --RESERVED FOR POWER MONITOR BIT FAILURE--
387
003564  000000                          0
003566  000000                          0
003570  000000                          0
003572  000000                          0

388
389                             ;ERROR  200     INCORRECT 'PIP'' STATUS DURING DIAGNOSTIC MODE
390
003574  072144                          EMT200
003576  074140                          EHT1
003600  074264                          EDT1
003602  074354                          EFT1

391
392                             ;ERROR  201     INCORRECT 'WRL'' STATUS DURING DIAGNOSTIC MODE
393
003604  072156                          EMT201
003606  074140                          EHT1
003610  074264                          EDT1
003612  074354                          EFT1

394
395                             ;ERROR  202     INCORRECT ''SKI'' STATUS DURING DIAGNOSTIC MODE
396
003614  072170                          EMT202
003616  074140                          EHT1
003620  074264                          EDT1
003622  074354                          EFT1

397
398                             ;ERROR  203     INCORRECT 'DVC'' STATUS DURING DIAGNOSTIC MODE
399
003624  072202                          EMT203
003626  074140                          EHT1
003630  074264                          EDT1
003632  074354                          EFT1

400
401                             ;ERROR  204     ''VV'' WAS NOT RESET BY MAINTENANCE UNIT READY
402
003634  072214                          EMT204
003636  074140                          EHT1
003640  074264                          EDT1
003642  074354                          EFT1

403
404                             ;ERROR  205     SELECTED DEVICE HAS A PERSISTENT ''SKI'' ERROR
405
003644  072232                          EMT205
003646  074140                          EHT1
003650  074264                          EDT1
```

ERROR POINTER TABLE

```
        003652  074354                          EFT1

    406
    407                             ;ERROR  206     'LBC'' DID NOT SET DURING DIAGNOSTIC MODE
    408
        003654  072242                          EMT206
        003656  074140                          EHT1
        003660  074264                          EDT1
        003662  074354                          EFT1

    409
    410                             ;ERROR  207     UNEXPECTED LOSS OF 'MOL'' - MEDIUM IS OFF LINE
    411
        003664  072252                          EMT207
        003666  074140                          EHT1
        003670  074264                          EDT1
        003672  074354                          EFT1

    412
    413                             ;ERROR  210     UNEXPECTED LOSS OF VOLUME VALID - 'VV'' = 0
    414
        003674  072264                          EMT210
        003676  074140                          EHT1
        003700  074264                          EDT1
        003702  074354                          EFT1

    415
    416                             ;ERROR  211     UNEXPECTED MECHANICAL MOTION - 'PIP'' = 1
    417
        003704  072272                          EMT211
        003706  074140                          EHT1
        003710  074264                          EDT1
        003712  074354                          EFT1

    418
    419                             ;ERROR  212     UNEXPECTED DEVICE FAULT - 'DVC'' = 1
    420
        003714  072306                          EMT212
        003716  074140                          EHT1
        003720  074264                          EDT1
        003722  074354                          EFT1

    421
    422                             ;ERROR  213     UNEXPECTED SEEK INCOMPLETE ERROR - 'SKI'' = 1
    423
        003724  072322                          EMT213
        003726  074140                          EHT1
        003730  074264                          EDT1
        003732  074354                          EFT1

    424
    425                             ;ERROR  214     DRIVE EXECUTED A RECALIBRATE WITH ERROR SET
    426
        003734  072332                          EMT214
        003736  074152                          EHT2
        003740  074274                          EDT2
        003742  074364                          EFT2
```

ERROR POINTER TABLE

```
                                ;ERROR  215     DRIVE DID NOT DETECT "IVC" ERROR DURING RECALIBRATE
003744  072352                          EMT215
003746  074152                          EHT2
003750  074274                          EDT2
003752  074364                          EFT2

                                ;ERROR  216     INCORRECT "IVC" STATUS
003754  072364                          EMT216
003756  074140                          EHT1
003760  074264                          EDT1
003762  074354                          EFT1

                                ;ERROR  217     INCORRECT "IAE" STATUS
003764  072374                          EMT217
003766  074140                          EHT1
003770  074264                          EDT1
003772  074354                          EFT1

                                ;ERROR 220      INCORRECT "WLE" STATUS
003774  072404                          EMT220
003776  074140                          EHT1
004000  074264                          EDT1
004002  074354                          EFT1

                                ;ERROR  221     INCORRECT "OPI" STATUS
004004  072414                          EMT221
004006  074140                          EHT1
004010  074264                          EDT1
004012  074354                          EFT1

                                ;ERROR  222     RM80 DID NOT DETECT RMR ERROR
004014  072424                          EMT222
004016  074140                          EHT1
004020  074264                          EDT1
004022  074354                          EFT1

                                ;ERROR  223     RM80 DI: NOT DETECT PARITY ERROR ON MASSBUS CONTROL BUS
004024  072434                          EMT223
004026  074176                          EHT223
004030  074310                          EDT223
004032  074400                          EFT223
```

ERROR POINTER TABLE

```
448
449                                     ;ERROR  224     SSE NOT DETECTED DURING DATA TRANSFER
4E^
     004034  072444                               EMT224
     004036  074140                               EHT1
     004040  074264                               EDT1
     004042  074354                               EFT1

451
452                                     ;ERROR  225     "SSEI" NOT RESET BY END OF TRACK DURING DATA TRANSFER
453
     004044  072460                               EMT225
     004046  074140                               EHT1
     004050  074264                               EDT1
     004052  074354                               EFT1

454
455                                     ;ERROR  226     "SSEI" SHOULD BE SET AFTER DATA TRANSFER
456
     004054  072500                               EMT226
     004056  074140                               EHT1
     004060  074264                               EDT1
     004062  074354                               EFT1

457
458                                     ;ERROR  227     SSE WAS DETECTED W/ SSEI SET
459
     004064  072510                               EMT227
     004066  074140                               EHT1
     004070  074264                               EDT1
     004072  074354                               EFT1

460
461                                     ;ERROR  230     UNUSED
462
     004074  072524                               EMT230
     004076  000000                               0
     004100  000000                               0
     004102  000000                               0

463
464                                     ;ERROR  231     UNUSED
465
     004104  072526                               EMT231
     004106  000000                               0
     004110  000000                               0
     004112  000000                               0

466
467                                     ;ERROR  232     UNUSED
468
     004114  072530                               EMT232
     004116  000000                               0
     004120  000000                               0
     004122  000000                               0

469
```

ERROR POINTER TABLE

```
470                                     ;ERROR  233     UNUSED
471
     004124  072532                               EMT233
     004126  000000                               0
     004130  000000                               0
     004132  000000                               0

472
473                                     ;ERROR  234     UNUSED
474
     004134  000000                               0
     004136  000000                               0
     004140  000000                               0
     004142  000000                               0

475
476                                     ;ERROR  235     UNUSED
477
     004144  000000                               0
     004146  000000                               0
     004150  000000                               0
     004152  000000                               0

478
479                                     ;ERROR  236     UNUSED
480
     004154  000000                               0
     004156  000000                               0
     004160  000000                               0
     004162  000000                               0

481
482                                     ;ERROR  237     UNUSED
483
     004164  000000                               0
     004166  000000                               0
     004170  000000                               0
     004172  000000                               0

484
485                                     ;ERROR  240     UNUSED
486
     004174  000000                               0
     004176  000000                               0
     004200  000000                               0
     004202  000000                               0

487
488                                     ;ERROR  241     UNUSED
489
     004204  000000                               0
     004206  000000                               0
     004210  000000                               0
     004212  000000                               0

490
491                                     ;ERROR  242     UNUSED
```

ERROR POINTER TABLE

```
492
    004214  000000                          0
    004216  000000                          0
    004220  000000                          0
    004222  000000                          0

493
494                                 ;ERROR  243     UNUSED
495
    004224  000000                          0
    004226  000000                          0
    004230  000000                          0
    004232  000000                          0

496
497                                 ;ERROR  244     UNUSED
498
    004234  000000                          0
    004236  000000                          0
    004240  000000                          0
    004242  000000                          0

499
500                                 ;ERROR  245     UNUSED
501
    004244  000000                          0
    004246  000000                          0
    004250  000000                          0
    004252  000000                          0

502
503                                 ;ERROR  246     "ATA" NOT RESET BY GO WHEN "ERR" = 0
504
    004254  072560                          EMT246
    004256  074140                          EHT1
    004260  074264                          EDT1
    004262  074354                          EFT1

505
506                                 ;ERROR  247     "ATA" NOT RESET BY WRITING RMAS
507
    004264  072570                          EMT247
    004266  074140                          EHT1
    004270  074264                          EDT1
    004272  074354                          EFT1

508
509                                 ;ERROR  250     "ATA" WAS RESET BY GO WHEN "ERR" = 1
510
    004274  072602                          EMT250
    004276  074140                          EHT1
    004300  074264                          EDT1
    004302  074354                          EFT1

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

ERROR POINTER TABLE

```
004304  072616                          EMT251
004306  074152                          EHT2
004310  074274                          EDT2
004312  074364                          EFT2

514
515                             ;ERROR  252     PROGRAM INTERRUPT SHOULD NOT HAVE BEEN GENERATED
516
004314  072624                          EMT252
004316  074152                          EHT2
004320  074274                          EDT2
004322  074364                          EFT2

517
518                             ;ERROR  253     OFFSET MODE WAS NOT RESET BY WRITING RMDC
519
004324  072632                          EMT253
004326  074140                          EHT1
004330  074264                          EDT1
004332  074354                          EFT1

520
521                             ;ERROR  254     INCORRECT "ILF" STATUS
522
004334  072650                          EMT254
004336  074140                          EHT1
004340  074264                          EDT1
004342  074354                          EFT1

523
524                             ;ERROR  255     INCORRECT "ATA" STATUS
525
004344  072660                          EMT255
004346  074140                          EHT1
004350  074264                          EDT1
004352  074354                          EFT1

526
527                             ;ERROR  256     INCORRECT "ILR" STATUS
528
004354  072670                          EMT256
004356  074210                          EHT256
004360  074310                          EDT223
004362  074400                          EFT223

529
530                             ;ERROR  257     INVALID IAE STATUS DURING SEARCH COMMAND
531
004364  072700                          EMT257
004366  074140                          EHT1
004370  074264                          EDT1
004372  074354                          EFT1

532
533                             ;ERROR  260     "IVC" WAS NOT DETECTED DURING SEARCH COMMAND
534
004374  072712                          EMT260
```

ERROR POINTER TABLE

```
004376  074140                          EHT1
004400  074264                          EDT1
004402  074354                          EFT1

535
536                             ;ERROR  261     DRIVE EXECUTED SEARCH WITH ERROR SET
537
004404  072724                          EMT261
004406  074140                          EHT1
004410  074264                          EDT1
004412  074354                          EFT1

538
539                             ;ERROR  262     "LBC" ERROR NOT SET DURING DIAGNOSTIC MODE
540
004414  072744                          EMT262
004416  074140                          EHT1
004420  074264                          EDT1
004422  074354                          EFT1

541
542                             ;ERROR  263     "SKI" ERROR DURING SEARCH COMMAND
543
004424  072754                          EMT263
004426  074140                          EHT1
004430  074264                          EDT1
004432  074354                          EFT1

544
545                             ;ERROR  264     "IVC" ERROR DURING SEARCH - LOST VOLUME VALID
546
004434  072764                          EMT264
004436  074140                          EHT1
004440  074264                          EDT1
004442  074354                          EFT1

547
548                             ;ERROR  265     ERRONEOUS IVC ERROR DURING SEARCH - VOLUME IS VALID
549
004444  073004                          EMT265
004446  074140                          EHT1
004450  074264                          EDT1
004452  074354                          EFT1

550
551                             ;ERROR  266     DEVICE FAULT (DVC) DURING SEARCH
552
004454  073024                          EMT266
004456  074140                          EHT1
004460  074264                          EDT1
004462  074354                          EFT1

553
554                             ;ERROR  267     SKI SHOULD HAVE BEEN SET BECAUSE CYLINDER
555                             ;               ADDRESS IS TOO LARGE
556
004464  073036                          EMT267
```

ERROR POINTER TABLE

```
004466  074140                          EHT1
004470  074264                          EDT1
004472  074354                          EFT1

557
558                             ;ERROR  270     OPI ERROR DURING SEARCH BECAUSE MOL = 0
559
004474  073054                          EMT270
004476  074140                          EHT1
004500  074264                          EDT1
004502  074354                          EFT1

560
561                             ;ERROR  271     OPI ERROR DURING SEARCH BECAUSE ON CYLINDER
562                             ;               DIDN'T DROP
563
004504  073070                          EMT271
004506  074140                          EHT1
004510  074264                          EDT1
004512  074354                          EFT1

564
565                             ;ERROR  272     LOST MOL DURING SEARCH, OPI IS NOT SET
566
004514  073106                          EMT272
004516  074140                          EHT1
004520  074264                          EDT1
004522  074354                          EFT1

567
568                             ;ERROR  273     PIP STIL SET AFTER SEARCH
569
004524  073124                          EMT273
004526  074140                          EHT1
004530  074264                          EDT1
004532  074354                          EFT1

570
571                             ;ERROR  274     PARITY ERROR OCCURRED WHILE WRITING REMOTE
572                             ;               REGISTERS BUT MXF DID NOT SET
573
004534  073142                          EMT274
004536  074140                          EHT1
004540  074264                          EDT1
004542  074354                          EFT1

574
575                             ;ERROR  275     MXF ERROR - COMPOSITE ERROR OCCURRED BEFORE DATA
576                             ;               COMMAND STARTED
577
004544  073160                          EMT275
004546  074140                          EHT1
004550  074264                          EDT1
004552  074354                          EFT1

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

ERROR POINTER TABLE

```
                                        ;               ZERO
004554  073172                                  EMT276
004556  074140                                  EHT1
004560  074264                                  EDT1
004562  074354                                  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
004564  073206                                  EMT277
004566  074140                                  EHT1
004570  074264                                  EDT1
004572  074354                                  EFT1

                                        ;ERROR  300     "IVC" ERROR DURING DATA TRANSFER BECAUSE VOLUME
                                        ;               WAS NOT VALID
004574  073224                                  EMT300
004576  074140                                  EHT1
004600  074264                                  EDT1
004602  074354                                  EFT1

                                        ;ERROR  301     ERRONEOUS "IVC" ERROR DURING DATA TRANSFER - VOLUME
                                        ;               IS VALID
004604  073244                                  EMT301
004606  074140                                  EHT1
004610  074264                                  EDT1
004612  074354                                  EFT1

                                        ;ERROR  302     "ILR" ERROR DURING DATA TRANSFER
004614  073266                                  EMT302
004616  074140                                  EHT1
004620  074264                                  EDT1
004622  074354                                  EFT1

                                        ;ERROR  303     "ILF" ERROR DURING DATA TRANSFER
004624  073300                                  EMT303
004626  074140                                  EHT1
004630  074264                                  EDT1
004632  074354                                  EFT1

                                        ;ERROR  304     "RMR" ERROR DURING DATA TRANSFER
004634  073312                                  EMT304
004636  074140                                  EHT1
004640  074264                                  EDT1
```

ERROR POINTER TABLE

```
        004642  074354                          EFT1

    604
    605                                 ;ERROR  305     INCORRECT ''IAE'' STATUS DURING DATA TRANSFER
    606
        004644  073324                          EMT305
        004646  074140                          EHT1
        004650  074264                          EDT1
        004652  074354                          EFT1

    607
    608                                 ;ERROR  306     ''SKI'' ERROR DURING DATA TRANSFER
    609
        004654  073336                          EMT306
        004656  074140                          EHT1
        004660  074264                          EDT1
        004662  074354                          EFT1

    610
    611                                 ;ERROR  307     DRIVE DID NOT DETECT SKI ERROR DUE TO CYLINDER
    612
        004664  073346                          EMT307
        004666  074140                          EHT1
        004670  074264                          EDT1
        004672  074354                          EFT1

    613
    614                                 ;ERROR  310     DEVICE FAULT DURING DATA TRANSFER
    615
        004674  073366                          EMT310
        004676  074140                          EHT1
        004700  074264                          EDT1
        004702  074354                          EFT1

    616
    617                                 ;ERROR  311     LOSS OF BIT CLOCK DURING DATA TRANSFER
    618
        004704  073400                          EMT311
        004706  074140                          EHT1
        004710  074264                          EDT1
        004712  074354                          EFT1

    619
    620                                 ;ERROR  312     LOSS OF SYSTEM CLOCK DURING DATA TRANSFER
    621
        004714  073412                          EMT312
        004716  074140                          EHT1
        004720  074264                          EDT1
        004722  074354                          EFT1

    622
    623                                 ;ERROR  313     UNSAFE ERROR DURING DATA TRANSFER (DVC = 0)
    624
        004724  073424                          EMT313
        004726  074140                          EHT1
        004730  074264                          EDT1
        004732  074354                          EFT1
```

```
                                ;ERROR  314     DRIVE TIMING ERROR DURING DATA TRANSFER

004734  073444                          EMT314
004736  074140                          EHT1
004740  074264                          EDT1
004742  074354                          EFT1

                                ;ERROR  315     WRITE LOCK ERROR

004744  073456                          EMT315
004746  074140                          EHT1
004750  074264                          EDT1
004752  074354                          EFT1

                                ;ERROR  316     ERRONEOUS WRITE LOCK ERROR

004754  073470                          EMT316
004756  074140                          EHT1
004760  074264                          EDT1
004762  074354                          EFT1

                                ;ERROR  317     HEADER CRC ERROR DURING DATA TRANSFER

004764  073502                          EMT317
004766  074140                          EHT1
004770  074264                          EDT1
004772  074354                          EFT1

                                ;ERROR  320     FORMAT ERROR DURING DATA TRANSFER

004774  073512                          EMT320
004776  074140                          EHT1
005000  074264                          EDT1
005002  074354                          EFT1

                                ;ERROR  321     HEADER COMPARE ERROR DURING DATA TRANSFER

005004  073522                          EMT321
005006  074140                          EHT1
005010  074264                          EDT1
005012  074354                          EFT1

                                ;ERROR  322     HEADER ERRORS SHOULD NOT BE SET

005014  073532                          EMT322
005016  074140                          EHT1
005020  074264                          EDT1
005022  074354                          EFT1
```

ERROR POINTER TABLE

```
646
647                                     ;ERROR  323     DATA CHECK ERROR DURING DATA TRANSFER
648
005024  073540                                  EMT323
005026  074140                                  EHT1
005030  074264                                  EDT1
005032  074354                                  EFT1

649
650                                     ;ERROR  324     CORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
651
005034  073550                                  EMT324
005036  074140                                  EHT1
005040  074264                                  EDT1
005042  074354                                  EFT1

652
653                                     ;ERROR  325     UNCORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
654
005044  073562                                  EMT325
005046  074140                                  EHT1
005050  074264                                  EDT1
005052  074354                                  EFT1

655
656                                     ;ERROR  326     DATA PARITY ERROR DURING READ COMMAND
657
005054  073574                                  EMT326
005056  074140                                  EHT1
005060  074264                                  EDT1
005062  074354                                  EFT1

658
659                                     ;ERROR  327     OFFSET MODE NOT RESET BY WRITE COMMAND
660
005064  073612                                  EMT327
005066  074140                                  EHT1
005070  074264                                  EDT1
005072  074354                                  EFT1

661
662                                     ;ERROR  330     DATA PARITY ERROR DURING WRITE COMMAND
663
005074  073624                                  EMT330
005076  074140                                  EHT1
005100  074264                                  EDT1
005102  074354                                  EFT1

664
665                                     ;ERROR  331     WRITE CLOCK FALURE DURING WRITE COMMAND
666
005104  073634                                  EMT331
005106  074140                                  EHT1
005110  074264                                  EDT1
005112  074354                                  EFT1

667
```

ERROR POINTER TABLE

```
668                                     ;ERROR  332     DATA LATE ERROR DURING DATA TRANSFER
669
    005114  073646                              EMT332
    005116  074140                              EHT1
    005120  074264                              EDT1
    005122  074354                              EFT1

670
671                                     ;ERROR  333     PIP STIL SET AFTER DATA TRANSFER - SKI = 0
672
    005124  073660                              EMT333
    005126  074140                              EHT1
    005130  074264                              EDT1
    005132  074354                              EFT1

673
674                                     ;ERROR  334     LOST MOL DURING DATA TRANSFER - OPI = 0
675
    005134  073676                              EMT334
    005136  074140                              EHT1
    005140  074264                              EDT1
    005142  074354                              EFT1

676
677                                     ;ERROR  335     LOST VOLUME VALID DURING DATA TRANSFER - IVC  = 0
678
    005144  073714                              EMT335
    005146  074140                              EHT1
    005150  074264                              EDT1
    005152  074354                              EFT1

679
680                                     ;ERROR  336     DATA READ DOES NOT COMPARE WITH DATA WRITTEN
681
    005154  073734                              EMT336
    005156  074222                              EHT336
    005160  074320                              EDT336
    005162  074410                              EFT336

682
683                                     ;ERROR  337     WRITE CHECK ERROR NOT DETECTED
684
    005164  073744                              EMT337
    005166  074234                              EHT337
    005170  074330                              EDT337
    005172  074420                              EFT337

685
686                                     ;ERROR  340     WRITE CHECK ERROR AT UNEXPECTED ADDRESS
687
    005174  073754                              EMT340
    005176  074222                              EHT336
    005200  074320                              EDT336
    005202  074410                              EFT336

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

ERROR POINTER TABLE

```
690
     005204  073766                          EMT341
     005206  074222                          EHT336
     005210  074320                          EDT336
     005212  074410                          EFT336

691
692                                 ;ERROR  342     "IVC" ERROR NOT DETECTED DURING DATA TRANSFER
693
     005214  073774                          EMT342
     005216  074140                          EHT1
     005220  074264                          EDT1
     005222  074354                          EFT1

694
695                                 ;ERROR  343     "FER" NOT DETECTED DURING DATA TRANSFER
696
     005224  074006                          EMT343
     005226  074140                          EHT1
     005230  074264                          EDT1
     005232  074354                          EFT1

697
698                                 ;ERROR  344     "HCE" NOT DETECTED DURING DATA TRANSFER
699
     005234  074020                          EMT344
     005236  074246                          EHT344
     005240  074340                          EDT344
     005242  074430                          EFT344

700
701                                 ;ERROR  345     "BSE" NOT DETECTED DURING DATA TRANSFER
702
     005244  074032                          EMT345
     005246  074140                          EHT1
     005250  074264                          EDT1
     005252  074354                          EFT1

703
704                                 ;ERROR  346     HEADER ERROR WAS DETECTED W/ HCI SET
705
     005254  074042                          EMT346
     005256  074140                          EHT1
     005260  074264                          EDT1
     005262  074354                          EFT1

706
707                                 ;ERROR  347     DATA TRANSFER NOT ABORTED W/ COMP ERROR SET
708
     005264  074056                          EMT347
     005266  074140                          EHT1
     005270  074264                          EDT1
     005272  074354                          EFT1

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

ERROR POINTER TABLE

```
005274  074070                          EMT350
005276  074140                          EHT1
005300  074264                          EDT1
005302  074354                          EFT1

712
713                             ;ERROR  351     "ATA" DID NOT SET DURING SEARCH
714
005304  074106                          EMT351
005306  074140                          EHT1
005310  074264                          EDT1
005312  074354                          EFT1

715
716                             ;ERROR  352     PROGRAM TIMEOUT WHILE TESTING RMLA
717
005314  074116                          EMT352
005316  000000                          0
005320  000000                          0
005322  000000                          0

718
719                             ;ERROR  353     LOOK AHEAD TEST FAILS
720
005324  074122                          EMT353
005326  074260                          EHT353
005330  074352                          EDT353
005332  074440                          EFT353

721
722                             ;ERROR  354     BSE SHOULD NOT BE SET
723
005334  074132                          EMT354
005336  074140                          EHT1
005340  074264                          EDT1
005342  074354                          EFT1

724
725                             ;PUT ERROR TABLE HERE
```

```
                        .SBTTL  ERROR TABLE USAGE

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

ERROR TABLE USAGE

```
                                        ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS

005344  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE THE TIME OUT OCCURED
005346  005740                          TST     -(R0)           ;ADJUST PC -2
005350  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
005352  104401  005360                  TYPE    ,65$            ;;TYPE ASCIZ STRING
005356  000417                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
005416                          64$:
005416  010046                          MOV     R0,-(SP)        ;SETUP FOR TYPING OUT PC
005420  104402                          TYPOC
005422  000240                          NOP                     ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
                                                                ;TO STOP ON UNEXPECTED TIMEOUT.


                                .SBTTL  START OF PROGRAM

005424  012737  177777  001330  START1: MOV     #-1,CHGADR      ;CHANGE RH/RM BUS ADDRESS
005432  000402                          BR      START2

005434  005037  001330          START:  CLR     CHGADR          ;NO CHANGE IN ADDRESS
005440  000240                  START2: NOP
005442  005227  000000                  INC     #0              ;TTY LOOP, WAIT FOR INCREMENT
005446  001375                          BNE     .-4             ;OF WORD
005450  000005                          RESET                   ;RESET THE WORLD


                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
005452  012706  001114                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
005456  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
005460  022706  001154                  CMP     #SWR,R6         ;;DONE?
005464  001374                          BNE     .-6             ;;LOOP BACK IF NO
005466  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
005472  012737  061470  000020          MOV     #$SCOPE,a#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005500  012737  000340  000022          MOV     #340,a#IOTVEC+2 ;;LEVEL 7
005506  012737  062250  000030          MOV     #$ERROR,a#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005514  012737  000340  000032          MOV     #340,a#EMTVEC+2 ;;LEVEL 7
005522  012737  065254  000034          MOV     #$TRAP,a#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005530  012737  000340  000036          MOV     #340,a#TRAPVEC+2;LEVEL 7
005536  012737  065362  000024          MOV     #$PWRDN,a#PWRVEC ;;POWER FAILURE VECTOR
005544  012737  000340  000026          MOV     #340,a#PWRVEC+2 ;;LEVEL 7
005552  013737  033614  033606          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
005560  005037  001206                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005564  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005570  112737  000001  001131          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005576  012737  005576  001122          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005604  012737  005604  001124          MOV     #.,$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.
005612  013746  000004                  MOV     a#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005616  012737  005652  000004          MOV     #64$,a#ERRVEC   ;;SET UP ERROR VECTOR
005624  012737  177570  001154          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005632  012737  177570  001156          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005640  022777  177777  173306          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005646  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005650  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
```

INITIALIZE THE COMMON TAGS

```
005652  012716  005660          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
005656  000002                          RTI
005660  012737  000176  001154  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
005666  012737  000174  001156          MOV     #DISPREG,DISPLAY
005674  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

005700  005037  001230                  CLR     $PASS           ;;CLEAR PASS COUNT
005704  132737  000200  001243          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
005712  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
005714  012737  001244  001154          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
005722                          67$:
                ;SETUP ''TIMEOUT'' TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
005722  012737  005344  000004          MOV     #BADTMO,ERRVEC  ;SETUP FOR UNEXPECTED TIMEOUT
005730  012737  000300  000006          MOV     #PR6,ERRVEC+2   ;LEVEL 6


                .SBTTL  TYPE PROGRAM NAME
                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
005736  005227  177777                  INC     #-1             ;;FIRST TIME?
005742  001033                          BNE     68$             ;;BRANCH IF NO
005744  022737  033756  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
005752  001427                          BEQ     68$             ;;BRANCH IF YES
005754  104401  005762                  TYPE    ,69$            ;;TYPE ASCIZ STRING
005760  000424                          BR      68$             ;;GET OVER THE ASCIZ
                        ;;69$:  .ASCIZ  <CRLF>@CZRNFA0 - RM80 FUNCTIONAL TEST, PT 3@<CRLF>
006032                          68$:
                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
006032  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
006036  001012                          BNE     70$             ;;BRANCH IF YES
006040  123727  001242  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
006046  001406                          BEQ     70$             ;;BRANCH IF YES
006050  023727  001154  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
006056  001005                          BNE     71$             ;;BRANCH IF NO
006060  104407                          GTSWR                   ;;GET SOFT-SWR SETTINGS
006062  000403                          BR      71$
006064  112737  000001  001150  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
006072                          71$:

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

006072  005037  001332                  CLR     XXDP            ;CLEAR 'XXDP' LOAD DEVICE STORAGE
006076  122737  000016  000041          CMPB    #16,@#41        ;LOADED FROM AN RM80 ?
006104  001121                          BNE     3$              ;BR IF NOT
006106  013737  000040  001332          MOV     @#40,XXDP       ;GET DEVICE INDICATOR AND NUMBER
006114  122737  000007  001332          CMPB    #7,XXDP         ;IS IT A VALID NUMBER ?
006122  103002                          BHIS    1$              ;YES
006124  105037  001332                  CLRB    XXDP            ;NO, DEFAULT TO DRIVE 0
006130  005737  000042          1$:     TST     @#42            ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
006134  001425                          BEQ     2$              ;BR IF NEITHER
006136  104401  006144                  TYPE    ,73$            ;;TYPE ASCIZ STRING
006142  000412                          BR      72$             ;;GET OVER THE ASCIZ
                        ;;73$:  .ASCIZ  <CRLF>/NOT TESTING DRIVE /
006170                          72$:
006170  005046                          CLR     -(SP)           ;CLEAR WORD ON STACK
006172  113716  001332                  MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
006176  104403                          TYPOS                   ;TYPE THE ADDRESS
006200     001                          .BYTE   1               ;ONLY 1 CHARACTER
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
                                                .BYTE   0               ;SUPRESS LEADING ZEROS
                                                TYPE    ,$CRLF          ;CR-LF
                                                BR      3$              ;GET NUMBER OF DRIVES

                                2$:             INC     #-1             ;FIRST TIME THRU HERE ?
                                                BNE     3$              ;NO
                                                TYPE    ,75$            ;;TYPE ASCIZ STRING
                                                BR      74$             ;;GET OVER THE ASCIZ
                                ;;75$:          .ASCIZ  <CRLF>/TO TEST DRIVE /
                                74$:
                                                CLR     -(SP)           ;CLEAR WORD ON STACK
                                                MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
                                                TYPOS                   ;TYPE DRIVE ADDRESS
                                                .BYTE   1               ;ONLY 1 CHARACTER
                                                .BYTE   0               ;SUPRESS LEADING ZEROS
                                                TYPE    ,76$            ;;TYPE ASCIZ STRING
                                                BR      3$              ;;GET OVER THE ASCIZ
                                ;;76$:          .ASCIZ  /, HALT PROGRAM, CLEAR LOC. 40 AND RESTART PROGRAM./<CRLF>
                                3$:

                                ;CHECK FOR AUTO MODE OR STANDALONE MODE
                                                TST     @#42            ;RUNNING IN AUTO MODE ?
                                                BEQ     STANDALONE      ;BR IF NO
                                                MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES

                                ;PROGRAM IS RUNNING IN AUTO MODE - SEE IF SIZING IS ALLOWED
                                XSIZ:
                                                BITB    #BIT7,$ENVM     ;SIZING ALLOWED ?
                                                BNE     12$             ;NO

                                                CLR     R1              ;START FROM DRIVE 0
                                                MOV     $BASE,R0        ;LOAD THE BASE ADDRESS
                                                TYPE    ,SYSTAT         ;TYPE 'UNIT STATUS:'

                                1$:             BITB    ATNTBL(R1),$DEVM        ;IS DEVICE PRESENT IN MAP ?
                                                BEQ     11$             ;BR IF NO
                                                TYPE    ,$CRLF          ;CR-LF
                                                MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
                                                TYPOS                   ;;GO TYPE--OCTAL ASCII
                                                .BYTE   2               ;;TYPE 2 DIGIT(S)
                                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
                                                TYPE    ,BLNKS4         ;TYPE 4 BLANKS

                                                MOV     #CLR,RMCS2(R0)  ;CLEAR MASS BUS
                                                MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
                                                TST     RMDS(R0)        ;ACCESS DRIVE REGISTER
                                                BIT     #NED,RMCS2(R0)  ;IS DRIVE PRESENT ?
                                                BNE     3$              ;BR IF NO
                                                BIT     #DVA,RMCS1(R0)  ;IS DRIVE AVAILABLE ?
                                                BEQ     4$              ;BR IF NO
                                                MOV     #$RM80,10$      ;ASSUME RM80 DEVICE
                                                CMP     #20026,RMDT(R0) ;SINGLE PORT RM80 ?
                                                BEQ     2$              ;BR IF YES
                                                CMP     #24026,RMDT(R0) ;DUAL PORT RM80 ?
                                                BEQ     2$              ;BR IF YES
                                                TYPE    ,NOTRM          ;DRIVE NOT AN RM80
                                                BR      5$              ;CHECK NEXT DRIVE
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
 98 006526  032760  010000  000012  2$:     BIT     #MOL,RMDS(R0)   ;IS MEDIUM ON LINE ?
 99 006534  001412                          BEQ     6$              ;BR IF NO
100 006536  000417                          BR      7$
101
102 006540  104401  067121          3$:     TYPE    ,NOTPRS         ;DRIVE NOT PRESENT
103 006544  000402                          BR      5$              ;CHECK NEXT DRIVE
104
105 006546  104401  067136          4$:     TYPE    ,NOTAVL         ;DRIVE NOT AVAILABLE
106 006552  146137  067344  001300  5$:     BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
107 006560  000425                          BR      11$             ;CHECK NEXT DRIVE
108
109 006562  104401  067155          6$:     TYPE    ,UNTOFF         ;DRIVE OFFLINE
110 006566  146137  067344  001300          BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
111 006574  000413                          BR      9$              ;PRINT DRIVE TYPE
112
113 006576  005737  001332          7$:     TST     XXDP            ;LOADED FROM RM80 ?
114 006602  001406                          BEQ     8$              ;NO
115 006604  123701  001332                  CMPB    XXDP,R1         ;IS THIS THE DRIVE ?
116 006610  001003                          BNE     8$              ;BR IF NO
117 006612  104401  067104                  TYPE    ,LODEV          ;DRIVE IS LOAD DEVICE
118 006616  000755                          BR      5$
119
120 006620  104401  067166          8$:     TYPE    ,UNTON          ;DRIVE ONLINE
121 006624  104401  067240          9$:     TYPE    ,BLNKS2         ;TYPE 2 BLANKS
122 006630  104401                          TYPE                    ;PRINT DRIVE TYPE
123 006632  000000                  10$:    .WORD   0               ;MESSAGE ADDRESS HERE
124
125 006634  005201                  11$:    INC     R1              ;INCREMENT THE DRIVE ADDRESS
126 006636  020127  000007                  CMP     R1,#7           ;ALL DRIVES ARE CHECKED ?
127 006642  003661                          BLE     1$              ;BRANCH IF NOT
128
129 006644  104401  001217          12$:    TYPE    ,$CRLF          ;CR-LF
130 006650  000137  007332                  JMP     CMNSTART        ;JUMP TO COMMON START
```

```
                                        .SBTTL  STANDALONE INPUT ROUTINES

006654                                  STANDALONE:
006654  004737  063670                          JSR     PC,$TKINT       ;INITIALIZE CONSOLE

006660  005227  177777                          INC     #-1             ;FIRST TIME THRU HERE ?
006664  001023                                  BNE     2$              ;BR IF NO

                                        ;SEE IF OPERATOR WANTS HELP TEXT

006666  104401  066110                          TYPE    ,MSHELP         ;WANT HELP ?
006672  104411                                  RDCHR                   ;GET RESPONSE
006674  0126    001176                        MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
006700  123727  001176  000131                  CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ?
006706  001005                                  BNE     1$              ;NO
006710  104401  001176                          TYPE    ,$TMP1          ;TYPE 'Y'
006714  104401  104254                          TYPE    ,HELP           ;YES - TYPE HELP TEXT
006720  000414                                  BR      3$
006722  104401  067232                1$:       TYPE    ,N              ;TYPE 'N'
006726  104401  001217                          TYPE    ,$CRLF          ;CR-LF
006732  000407                                  BR      3$

                                        ;SEE IF USER WANTS TO CHANGE UNIBUS ADDRESS
006734                                  2$:
006734  005737  001330                          TST     CHGADR          ;CHANGE RH/RM BUS ADDRESS ?
006740  001457                                  BEQ     7$              ;BR IF NO
006742  005037  001330                          CLR     CHGADR          ;NO CHANGE NEXT TIME
006746  104401  001217                          TYPE    ,$CRLF          ;CR-LF

                                        ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, VECTOR ADDRESS AND INTERRUPT PRIORITY
006752                                  3$:
006752  104401  066521                          TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
006756  013746  001276                          MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
006762  104402                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
006764  104401  067240                          TYPE    ,BLNKS2         ;TYPE 2 BLANKS
006770  104413                                  RDOCT                   ;GET NEW BUS ADDRESS
006772  012637  001176                          MOV     (SP)+,$TMP1     ;CARRIAGE RETURN ?
006776  001412                                  BEQ     5$              ;YES-SKIP TO NEXT ENTRY
007000  022737  160000  001176                  CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE ?
007006  101403                                  BLOS    4$              ;YES
007010  104401  066531                          TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
007014  000756                                  BR      3$              ;TRY AGAIN
007016  013737  001176  001276        4$:       MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS

007024  104401  066573                5$:       TYPE    ,CNSL03
007030  005046                                  CLR     -(SP)
007032  113716  001272                          MOVB    $VECT1,(SP)     ;GET CURRENT VECTOR ADDRESS
007036  104402                                  TYPOC
007040  104401  067240                          TYPE    ,BLNKS2         ;TYPE 2 BLANKS
007044  104413                                  RDOCT                   ;GET NEW VECTOR ADDRESS
007046  012637  001176                          MOV     (SP)+,$TMP1     ;CARRIAGE RETURN?
007052  001412                                  BEQ     7$              ;YES-SKIP TO NEXT ENTRY
007054  022737  001000  001176                  CMP     #1000,$TMP1     ;VECTOR ADDRESS < 1000 ?
007062  101003                                  BHI     6$              ;YES!!
007064  104401  066602                          TYPE    ,CNSL04         ;TYPE WARNING MESSAGE
007070  000755                                  BR      5$              ;RETRY
007072  113737  001176  001272        6$:       MOVB    $TMP1,$VECT1    ;STORE NEW VECTOR ADDRESS
```

STANDALONE INPUT ROUTINES

```
                                        ;DIALOGUE TO INPUT DEVICE NUMBERS
007100  005227  177777          7$:     INC     #-1             ;FIRST TIME THRU ?
007104  001002                          BNE     8$              ;BR IF NO
007106  104401  066636                  TYPE    ,CNSL07         ;TYPE INPUT INSTRUCTIONS
007112  104401  001217          8$:     TYPE    ,$CRLF          ;CR-LF
007116  005037  001300          9$:     CLR     $DEVM           ;CLEAR DEVICE MAP
007122  104401  067022                  TYPE    ,MSDRVS         ;TYPE 'DRIVE(S): '
007126  104411                          RDCHR
007130  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
007134  023727  001176  000101          CMP     $TMP1,#'A       ;IS INPUT "A" ?
007142  001007                          BNE     10$             ;NO
007144  104401  066074                  TYPE    ,ALL            ;YES, TYPE "ALL" AND GO
007150  012737  000377  001300          MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
007156  000137  006364                  JMP     XSIZ            ;AUTO SIZE.

007162  023727  001176  000015  10$:     CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
007170  001436                          BEQ     12$             ;YES
007172  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
007176  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
007204  002430                          BLT     12$             ;YES
007206  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
007214  003427                          BLE     13$             ;NO
007216  000423                          BR      12$             ;ILLEGAL INPUT

007220  104411          11$:    RDCHR
007222  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
007226  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
007234  001432                          BEQ     14$             ;YES
007236  104401  066105                  TYPE    ,COMMA          ;TYPE ', '
007242  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
007246  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
007254  002404                          BLT     12$             ;YES
007256  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
007264  003403                          BLE     13$             ;NO
007266  104401  067000          12$:    TYPE    ,CNSL08         ;TYPE " ?ILLEGAL INPUT"
007272  000711                          BR      9$              ;RETRY

007274  013701  001176          13$:    MOV     $TMP1,R1        ;R1 = DRIVE NUMBER
007300  042701  177770                  BIC     #^C7,R1
007304  156137  067344  001300          BISB    ATNTBL(R1),$DEVM        ;SET DEVICE IN MAP
007312  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ?
007320  101337                          BHI     11$             ;NO
007322  104401  001217          14$:    TYPE    ,$CRLF          ;CR-LF
007326  000137  006364                  JMP     XSIZ            ;GO SIZE DEVICES
```

```
                                ;ASSEMBLE TEST QUE FROM DEVICE MAP
                                CMNSTART:
                                        TYPE    ,DRIVES         ;TYPE 'DRIVE(S) TO BE TESTED'
                                        MOV     $DEVM,R0        ;R0 = DEVICE MAP
                                        BNE     1$              ;BR IF DRIVES TO TEST
                                        TYPE    ,COMMA          ;TYPE ', '
                                        TYPE    ,NONE           ;TYPE 'NONE'
                                1$:     MOV     #TSTQUE+2,R1    ;R1 = ADDRESS OF FIRST ENTRY IN QUE
                                        MOV     R1,TSTQUE       ;INITIALIZE ENTRY POINTER
                                        MOV     #1,R2           ;R2 = DEVICE POINTER
                                        CLR     R3              ;R3 = DEVICE NUMBER
                                2$:     BIT     R2,R0           ;IS THIS DEVICE IN MAP ?
                                        BEQ     3$              ;NO !!
                                        TYPE    ,COMMA          ;TYPE ', '
                                        MOV     R3,(R1)         ;YES - ENTER DEVICE NUMBER IN QUE
                                        MOV     R3,-(SP)        ;;SAVE R3 FOR TYPEOUT
                                        TYPOS                   ;;GO TYPE--OCTAL ASCII
                                        .BYTE   1               ;;TYPE 1 DIGIT(S)
                                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
                                        MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
                                        ADD     #2,R1           ;ADVANCE ENTRY POINTER
                                3$:     ASL     R2              ;ADVANCE DEVICE POINTER
                                        TSTB    R2              ;DONE ALL DEVICES ?
                                        BEQ     4$              ;YES
                                        INC     R3              ;ADVANCE DEVICE NUMBER
                                        BR      2$              ;ENTER NEXT DEVICE
                                4$:     CLR     (R1)            ;TERMINATE TEST QUE
                                        TYPE    ,$CRLF          ;CR-LF

                                ;SIZE FOR CLOCK
                                        JSR     PC,SIZCLK       ;SEE IF CLOCK PRESENT
                                        BR      6$              ;YES - CLOCK IS PRESENT
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/NO 'L' OR 'P' CLOCK/
                                64$:
                                        TST     @#42            ;ANY MONITOR PRESENT ?
                                        BNE     5$              ;BR IF YES
                                        JMP     START           ;JUMP TO START
                                5$:     JMP     $GET42          ;RETURN CONTROL TO MONITOR
                                6$:

                                READY:  NOP                     ;READY TO START TEST
                                        TSTB    $DEVM           ;ANY DRIVES IN MAP ?
                                        BNE     2$              ;BR IF YES
                                        TST     @#42            ;ANY MONITOR PRESENT ?
                                        BNE     1$              ;BR IF YES
                                        JMP     START           ;JUMP TO START
                                1$:     JMP     $GET42          ;RETURN CONTROL TO MONITOR

                                2$:     CLRB    $TSTNM          ;RESET TEST NUMBER
                                        CLR     $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
                                        CLR     CTLFG           ;CLEAR CONTROL-C FLAG
                                        CLR     MEDENB          ;CLEAR MEDIA ENABLE
                                        JSR     PC,$TKINT       ;INITIALIZE TTY
                                        MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
                                        MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
```

```
                007606  000002                          RTI                     ;;POP NEW PC AND PS
                007610                          64$:
     51         007610  117737  171652  001234          MOVB    @TSTQUE,$UNIT   ;LOAD DRIVE NUMBER
     52
     53                                         ;CLEAR MASSBUS CONTROLLER, SELECT DRIVE AND SET LAST TRACK ADDRESS
     54         007616  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     55         007622  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
     56         007630  113760  001234  000010          MOVB    $UNIT,RMCS2(R0) ;SELECT DEVICE UNDER TEST
     57         007636  012737  006400  001334          MOV     #TA8!TA4!TA1,LSTRK      ;SET LAST TRACK = 13.
     58
     59                                         ;TYPE DRIVE NUMBER TO BE TESTED($UNIT)
     60         007644  104401  001217                  TYPE    ,$CRLF          ;CR-LF
     61         007650  104401  067036                  TYPE    ,MSGDRV         ;TYPE 'DRIVE'
     62         007654  013746  001234                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                                 ;;TYPE DRIVE NUMBER
                007660  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
                007662     002                          .BYTE   2               ;;TYPE 2 DIGIT(S)
                007663     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
     63         007664  005004                          CLR     R4              ;THESE TWO LOOPS ARE ADDED TO
     64         007666  005304                          DEC     R4              ;WAIT FOR TTY
     65         007670  001376                          BNE     .-2
     66         007672  005304                          DEC     R4
     67         007674  001376                          BNE     .-2
```

```
                                ;;*****************************************************************
                                ;*TEST 1         CONTROLLER ACCESS TEST
                                ;;***************** *************************************************
                                TST1:
007676                                  SCOPE                           ;SCOPE CALL
007676  000004                         NOP                             ;START OF TEST
007700  000240                         MOV     #STACK,SP               ;INITIALIZE STACK POINTER
007702  012706  001100                 MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
007706  013700  001276                 MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
007712  013701  001466                 MOV     #1,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX
007716  012737  000001  001226
                                
007724  005001                         CLR     R1
007726  013746  000004                 MOV     ERRVEC,-(SP)            ;;PUSH ERRVEC ON STACK
007732  013746  000006                 MOV     ERRVEC+2,-(SP)                  ;;PUSH ERRVEC+2 ON STACK
007736  012737  010040  000004         MOV     #1$,ERRVEC
007744  012737  000300  000006         MOV     #PR6,ERRVEC+2
                                
007752  110160  000001                 MOVB    R1,RMCS1+1(R0)          ;MOVE HI BYTE TO RMCS1
007756  010160  000002                 MOV     R1,RMWC(R0)             ;MOVE WORD COUNT REGISTER
007762  016002  000002                 MOV     RMWC(R0),R2
007766  010160  000004                 MOV     R1,RMBA(R0)             ;MOVE BUS ADDRESS REGISTER
007772  016002  000004                 MOV     RMBA(R0),R2
007776  016046  000010                 MOV     RMCS2(R0),-(SP)                 ;;PUSH RMCS2(R0) ON STACK
010002  010160  000010                 MOV     R1,RMCS2(R0)            ;MOVE CONTROL STATUS REGISTER
010006  016002  000010                 MOV     RMCS2(R0),R2
010012  012660  000010                 MOV     (SP)+,RMCS2(R0)                 ;;POP STACK INTO RMCS2(R0)
010016  010160  000022                 MOV     R1,RMDB(R0)             ;MOVE DATA BUFFER
010022  016002  000022                 MOV     RMDB(R0),R2
010026  012637  000006                 MOV     (SP)+,ERRVEC+2                  ;;POP STACK INTO ERRVEC+2
010032  012637  000004                 MOV     (SP)+,ERRVEC            ;;POP STACK INTO ERRVEC
010036  000417                         BR      3$                      ;NO BUS TIMEOUT OCCURRED
                                
010040  022626                 1$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
010042  012637  000006                 MOV     (SP)+,ERRVEC+2                  ;;POP STACK INTO ERRVEC+2
010046  012637  000004                 MOV     (SP)+,ERRVEC            ;;POP STACK INTO ERRVEC
010052  104110                         EMT     110
010054  005737  000042                 TST     @#42                    ;STAND ALONE MODE ?
010060  001002                         BNE     2$                      ;NO!!
010062  000137  005424                 JMP     START1                  ;YES-GO GET $BASE
010066  005037  001300         2$:     CLR     $DEVM                   ;FUDGE NO DRIVES IN MAP
010072  000137  033746                 JMP     $GET42                  ;RETURN CONTROL TO MONITOR
010076                           3$:
                                
                                ;;*****************************************************************
                                ;*TEST 2         WRITE, READ ZEROS
                                ;;*****************************************************************
                                TST2:
010076                                  SCOPE                           ;SCOPE CALL
010076  000004                         NOP                             ;START OF TEST
010100  000240                         MOV     #STACK,SP               ;INITIALIZE STACK POINTER
010102  012706  001100                 MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
010106  013700  001276                 MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
010112  013701  001466                 MOV     #2,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX
010116  012737  000002  001226
                                
                                ;;*****************************************************************
                                ;LOOP #1         FORMAT,WRITE,READ
```

```
                                1$:
                                ;PREPARE DEVICE FOR TEST
010124  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
010130  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                            ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                            ;VERIFY PACK ACKNOWLEDGE
                                                            ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                            ;VERIFY RECALIBRATION
010132  000404                      BR      2$              ;GO TO 2$ IF NO ERROR
010134  000240                      NOP                     ;RETURN HERE IF ERROR
010136  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
010140  000137  011104              JMP     25$             ;GO TO 25$ IF ERROR

                                ;LOAD PARAMETERS AND GENERATE DATA BUFFER
                                2$:
010144  012737  001057  001446      MOV     #559.,RMDC0             ;CYLINDER = 559.
010152  012737  001000  001420      MOV     #TA2,RMDA0              ;TRACK = 2, SECTOR = 0
010160  012737  104254  001416      MOV     #BUFONE,RMBA0           ;BUS ADDRESS
010166  012737  177376  001414      MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
010174  012737  010000  001444      MOV     #FMT16,RMOF0            ;16 BIT FORMAT
010202  012737  000063  001412      MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND

                                ;VERIFY THAT SECTOR IS NOT BAD
010210  004737  034762              JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
010214  000405                      BR      3$              ;GO TO 3$ IF NO ERROR
010216  104401  066006              TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
010222  104000                      EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
010224  000137  011104              JMP     25$             ;GO TO 25$ IF ERROR
010230                          3$:
010230  012737  067456  001174      MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
010236  012737  000001  001176      MOV     #1,$TMP1                ;RANGE OF PATTERN
010244  004737  037214              JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                ;SETUP PARAMETERS AND EXCUTE COMMAND
010250  012702  001555              MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
010254  112722  000034              MOVB    #RMDC,(R2)+
010260  112722  000006              MOVB    #RMDA,(R2)+
010264  112722  000004              MOVB    #RMBA,(R2)+
010270  112722  000002              MOVB    #RMWC,(R2)+
010274  112722  000032              MOVB    #RMOF,(R2)+
010300  112722  000000              MOVB    #RMCS1,(R2)+
010304  112712  000200              MOVB    #200,(R2)               ;TERMINATE TABLE
                                4$:
010310  004737  040360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
010314  000404                      BR      5$              ;GO TO 5$ IF NO ERROR
010316  000240                      NOP                     ;RETURN HERE IF ERROR
010320  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
010322  000137  011104              JMP     25$             ;GO TO 25$ IF ERROR
010326                          5$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
010326  004737  040024              JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
```

```
                                        ;WAIT FOR COMMAND TO COMPLETE
 010332  004737  040732                         JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
 010336  004737  040110                         JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
 010342  000404                                 BR      6$              ;GO TO 6$ IF NO ERROR
 010344  000240                                 NOP                     ;RETURN HERE IF ERROR
 010346  104000                                 EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
 010350  000137  011104                         JMP     25$             ;GO TO 25$ IF ERROR
 010354                                 6$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
 010354  004737  054166                         JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
 010360  000405                                 BR      7$              ;GO TO 7$ IF NO ERROR
 010362  000240                                 NOP                     ;RETURN HERE IF ERROR
 010364  104000                                 EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
 010366  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 010370  000137  011104                         JMP     25$             ;GO TO 25$ IF ERROR
 010374                                 7$:

                                        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 010374  012737  010404  001124                 MOV     #8$,$LPERR
 010402  000410                                 BR      9$                      ;SKIP TO WRITE OPERATION

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

 010404                                 8$:

                                        ;PREPARE DEVICE FOR TEST
 010404  004737  034032                 JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
 010410  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
 010412  000404                         BR      9$              ;GO TO 9$ IF NO ERROR
 010414  000240                         NOP                     ;RETURN HERE IF ERROR
 010416  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
 010420  000137  011104                 JMP     25$             ;GO TO 25$ IF ERROR
 010424                         9$:

                                ;SETUP PARAMETERS AND EXCUTE COMMAND
 010424                         10$:
 010424  012737  104260  001416         MOV     #BUFONE+4,RMBAO         ;MOVE MEMORY ADDRESS
 010432  012737  177400  001414         MOV     #-256.,RMWCO            ;CHANGE WORD COUNT
 010440  012737  000061  001412         MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
 010446  012702  001555                 MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 010452  112722  000006                 MOVB    #RMDA,(R2)+
 010456  112722  000034                 MOVB    #RMDC,(R2)+
 010462  112722  000032                 MOVB    #RMCF,(R2)+
 010466  112722  000004                 MOVB    #RMBA,(R2)+
 010472  112722  000002                 MOVB    #RMWC,(R2)+
```

```
    107 010476  112722  000000                  MOVB    #RMCS1,(R2)+
    108 010502  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
    109
    110 010506  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        010512  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
        010514  000240                          NOP                     ;RETURN HERE IF ERROR
        010516  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        010520  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    111 010524                          11$:
    112
    113                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        010524  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    114
    115                                 ;WAIT FOR COMMAND TO COMPLETE
        010530  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    116
    117                                 ;GO GET REGISTER STATUS
    118 010534  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        010540  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
        010542  000240                          NOP                     ;RETURN HERE IF ERROR
        010544  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        010546  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    119 010552                          12$:
    120
    121                                 ;VERIFY RESULTS OF WRITE COMMAND
    122 010552  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        010556  000405                          BR      13$             ;GO TO 13$ IF NO ERROR
        010560  000240                          NOP                     ;RETURN HERE IF ERROR
        010562  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        010564  004736                          JSP     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        010566  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    123 010572                          13$:
    124 010572  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        010576  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
        010600  000240                          NOP                     ;RETURN HERE IF ERROR
        010602  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        010604  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        010606  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    125 010612                          14$:
    126 010612  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        010616  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
        010620  000240                          NOP                     ;RETURN HERE IF ERROR
        010622  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        010624  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        010626  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    127 010632                          15$:
    128
    129                                 ;CHANGE LOOP ADDRESSES
    130 010632  012737  010642  001124          MOV     #16$,$LPERR
    131 010640  000410                          BR      17$                     ;SKIP TO NEXT OPERATION
    132
    133                                 ;;*****************************************************************
    134                                 ;LOOP #3        READ
    135
    136 010642                          16$:
    137
    138                                 ;PREPARE DEVICE FOR TEST
```

```
    139 010642  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        010646  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
        010650  000404                          BR      17$             ;GO TO 17$ IF NO ERROR
        010652  000240                          NOP                     ;RETURN HERE IF ERROR
        010654  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        010656  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    140 010662                          17$:
    141
    142                                 ;SETUP PARAMETERS AND EXCUTE COMMAND
    143 010662                          18$:
    144 010662  012737  105264  001416          MOV     #BUFTWO+4,RMBAO         ;CHANGE MEMORY ADDRESS
    145 010670  012737  000071  001412          MOV     #RD!GO,RMCS1O           ;READ DATA COMMAND
    146 010676  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    147 010702  112722  000006                  MOVB    #RMDA,(R2)+
    148 010706  112722  000032                  MOVB    #RMOF,(R2)+
    149 010712  112722  000034                  MOVB    #RMDC,(R2)+
    150 010716  112722  000004                  MOVB    #RMBA,(R2)+
    151 010722  112722  000002                  MOVB    #RMWC,(R2)+
    152 010726  112722  000000                  MOVB    #RMCS1,(R2)+
    153 010732  112712  000200                  MOVB    #200,(R2)
    154
    155 010736  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        010742  000404                          BR      19$             ;GO TO 19$ IF NO ERROR
        010744  000240                          NOP                     ;RETURN HERE IF ERROR
        010746  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        010750  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    156 010754                          19$:
    157
    158                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        010754  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    159
    160                                 ;WAIT FOR COMMAND TO COMPLETE
        010760  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    161
    162                                 ;GO GET REGISTER STATUS
    163 010764  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        010770  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        010772  000240                          NOP                     ;RETURN HERE IF ERROR
        010774  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        010776  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    164 011002                          20$:
    165
    166                                 ;VERIFY RESULTS OF READ COMMAND
    167 011002  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        011006  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
        011010  000240                          NOP                     ;RETURN HERE IF ERROR
        011012  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        011014  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011016  000137  011104                  JMP     25$             ;GO TO 25$ IF ERROR
    168 011022                          21$:
```

T2 WRITE, READ ZEROS

```
    169 011022  004737  054166              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        011026  000405                      BR      22$             ;GO TO 22$ IF NO ERROR
        011030  000240                      NOP                     ;RETURN HERE IF ERROR
        011032  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        011034  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011036  000137  011104              JMP     25$             ;GO TO 25$ IF ERROR
    170 011042                          22$:
    171 011042  004737  041760              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        011046  000405                      BR      23$             ;GO TO 23$ IF NO ERROR
        011050  000240                      NOP                     ;RETURN HERE IF ERROR
        011052  104000                      EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        011054  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        011056  000137  011104              JMP     25$             ;GO TO 25$ IF ERROR
    172 011062                          23$:
    173
    174                                 ;GO VERIFY DATA
    175 011062  004737  037462              JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        011066  104260                      .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        011070  105264                      .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        011072  000404                      BR      24$             ;GO TO 24$ IF NO ERROR
        011074  000240                      NOP                     ;RETURN HERE IF ERROR
        011076  104000                      EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        011100  000137  011104              JMP     25$             ;GO TO 25$ IF ERROR
    176 011104                          24$:
    177
    178 011104                          25$:
    179
    180                                 ;;*****************************************************************
                                        ;*TEST 3         WRITE, WRITE CHECK ZEROS
                                        ;;*****************************************************************
        011104                          TST3:
        011104  000004                      SCOPE                   ;SCOPE CALL
        011106  000240                      NOP                     ;START OF TEST
        011110  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        011114  013700  001276              MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        011120  013701  001466              MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        011124  012737  000003  001226      MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
    181
    182                                 ;;*****************************************************************
    183                                 ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
    184
    185 011132                          1$:
    186
    187                                 ;PREPARE DEVICE FOR TEST
    188 011132  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        011136  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
        011140  000404                      BR      2$              ;GO TO 2$ IF NO ERROR
        011142  000240                      NOP                     ;RETURN HERE IF ERROR
        011144  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        011146  000137  012062              JMP     24$             ;GO TO 24$ IF ERROR
```

```
                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
011152                                  2$:
011152  012737  001057  001446                MOV     #559.,RMDCO             ;CYLINDER = 559.
011160  012737  001000  001420                MOV     #TA2,RMDAO              ;TRACK = 2, SECTOR = 0
011166  012737  104254  001416                MOV     #BUFONE,RMBAO           ;BUS ADDRESS
011174  012737  177376  001414                MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
011202  012737  010000  001444                MOV     #FMT16,RMOFO            ;16 BIT FORMAT
011210  012737  000063  001412                MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND

                                        ;VERIFY THAT SECTOR IS NOT BAD
011216  00473,  034762                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
011222  000405                              BR      3$              ;GO TO 3$ IF NO ERROR
011224  1044'   066006                      TYPE    .SCTMSG         ;TYPE BAD SECTOR MESSAGE
011230  104000                              EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
011232  000137  012062                      JMP     24$             ;GO TO 24$ IF ERROR
011236                                  3$:
011236  012737  067456  001174                MOV     #ZEROS,$TMPO            ;STARTING ADDRESS OF PATTERN
011244  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
011252  004737  037214                      JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
011256  012702  001555                      MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
011262  112722  000034                      MOVB    #RMDC,(R2)+
011266  112722  000006                      MOVB    #RMDA,(R2)+
011272  112722  000004                      MOVB    #RMBA,(R2)+
011276  112722  000002                      MOVB    #RMWC,(R2)+
011302  112722  000032                      MOVB    #RMOF,(R2)+
011306  112722  000000                      MOVB    #RMCS1,(R2)+
011312  112712  000200                      MOVB    #200,(R2)               ;TERMINATE TABLE
011316                                  4$:
011316  004737  040360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011322  000404                              BR      5$              ;GO TO 5$ IF NO ERROR
011324  000240                              NOP                     ;RETURN HERE IF ERROR
011326  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
011330  000137  012062                      JMP     24$             ;GO TO 24$ IF ERROR
011334                                  5$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
011334  004737  040024                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
011340  004737  040732                      JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO GET REGISTER STATUS
011344  004737  040110                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
011350  000404                              BR      6$              ;GO TO 6$ IF NO ERROR
011352  000240                              NOP                     ;RETURN HERE IF ERROR
011354  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
011356  000137  012062                      JMP     24$             ;GO TO 24$ IF ERROR
011362                                  6$:

                                        ;VERIFY RESULTS OF WRITE COMMAND
011362  004737  054166                      JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
011366  000405                              BR      7$              ;GO TO 7$ IF NO ERROR
011370  000240                              NOP                     ;RETURN HERE IF ERROR
011372  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
```

```
011374  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011376  000137  012062                   JMP     24$             ;GO TO 24$ IF ERROR
228 011402                      7$:
229
230                             ;MOVE LOOP ADDRESSES TO NEXT OPERATION
231 011402  012737  011412  001124       MOV     #8$,$LPERR
232 011410  000410                      BR      9$                      ;SKIP TO WRITE OPERATION
233
234                             ;;*****************************************************************
235                             ;LOOP #2        WRITE,WRITE CHECK DATA
236
237 011412                      8$:
238
239                             ;PREPARE DEVICE FOR TEST
240 011412  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    011416  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
    011420  000404                      BR      9$              ;GO TO 9$ IF NO ERROR
    011422  000240                      NOP                     ;RETURN HERE IF ERROR
    011424  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    011426  000137  012062              JMP     24$             ;GO TO 24$ IF ERROR
241 011432                      9$:
242
243                             ;SETUP PARAMETERS AND EXCUTE COMMAND
244 011432                      10$:
245 011432  012737  104260  001416          MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
246 011440  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
247 011446  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
248 011454  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
249 011460  112722  000006                  MOVB    #RMDA,(R2)+
250 011464  112722  000034                  MOVB    #RMDC,(R2)+
251 011470  112722  000032                  MOVB    #RMOF,(R2)+
252 011474  112722  000004                  MOVB    #RMBA,(R2)+
253 011500  112722  000002                  MOVB    #RMWC,(R2)+
254 011504  112722  000000                  MOVB    #RMCS1,(R2)+
255 011510  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
256
257 011514  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    011520  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
    011522  000240                          NOP                     ;RETURN HERE IF ERROR
    011524  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    011526  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
258 011532                      11$:
259
260                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    011532  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
261
262                             ;WAIT FOR COMMAND TO COMPLETE
    011536  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
263
264                             ;GO GET REGISTER STATUS
```

```
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
011546  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
011550  000240                          NOP                     ;RETURN HERE IF ERROR
011552  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
011554  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
011560                          12$:

                                ;VERIFY RESULTS OF WRITE COMMAND
011560  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
011564  000405                          BR      13$             ;GO TO 13$ IF NO ERROR
011566  000240                          NOP                     ;RETURN HERE IF ERROR
011570  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
011572  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011574  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
011600                          13$:
011600  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
011604  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
011606  000240                          NOP                     ;RETURN HERE IF ERROR
011610  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
011612  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011614  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
011620                          14$:
011620  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
011624  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
011626  000240                          NOP                     ;RETURN HERE IF ERROR
011630  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
011632  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
011634  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
011640                          15$:

                                ;CHANGE LOOP ADDRESSES
011640  012737  011650  001124          MOV     #16$,$LPERR
011646  000410                          BR      17$                     ;SKIP TO NEXT OPERATION

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

011650                          16$:

                                ;PREPARE DEVICE FOR TEST
011650  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
011654  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAI'ABLE
                                                                ;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
011656  000404                          BR      17$             ;GO TO 17$ IF NO ERROR
011660  000240                          NOP                     ;RETURN HERE IF ERROR
011662  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
011664  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
011670                          17$:

                                ;SETUP PARAMETERS AND EXCUTE COMMAND
011670                          18$:
```

```
  291 011670  012737  000051  001412          MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
  292 011676  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  293 011702  112722  000006                  MOVB    #PMDA,(R2)+
  294 011706  112722  000032                  MOVB    #RMOF,(R2)+
  295 011712  112722  000034                  MOVB    #RMDC,(R2)+
  296 011716  112722  000004                  MOVB    #RMBA,(R2)+
  297 011722  112722  000002                  MOVB    #RMWC,(R2)+
  298 011726  112722  000000                  MOVB    #RMCS1,(R2)+
  299 011732  112712  000200                  MOVB    #200,(R2)
  300
  301 011736  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      011742  000404                          BR      19$             ;GO TO 19$ IF NO ERROR
      011744  000240                          NOP                     ;RETURN HERE IF ERROR
      011746  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      011750  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
  302 011754                          19$:
  303
  304                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      011754  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  305
  306                                 ;WAIT FOR COMMAND TO COMPLETE
      011760  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  307
  308                                 ;GO GET REGISTER STATUS
  309 011764  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      011770  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
      011772  000240                          NOP                     ;RETURN HERE IF ERROR
      011774  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      011776  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
  310 012002                          20$:
  311
  312                                 ;VERIFY RESULTS OF WRITE CHECK COMMAND
  313 012002  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      012006  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
      012010  000240                          NOP                     ;RETURN HERE IF ERROR
      012012  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      012014  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      012016  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
  314 012022                          21$:
  315 012022  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      012026  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
      012030  000240                          NOP                     ;RETURN HERE IF ERROR
      012032  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      012034  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      012036  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
  316 012042                          22$:
  317 012042  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      012046  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
      012050  000240                          NOP                     ;RETURN HERE IF ERROR
      012052  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      012054  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      012056  000137  012062                  JMP     24$             ;GO TO 24$ IF ERROR
  318 012062                          23$:
  319
  320 012062                          24$:
  321
  322                                 ;;*****************************************************************
```

```
                                        ;*TEST 4        WRITE, WRITE CHECK ZEROS W/ WCE ERROR
                                        ;;*****************************************************************
        012062                          TST4:
        012062  000004                          SCOPE                   ;SCOPE CALL
        012064  000240                          NOP                     ;START OF TEST
        012066  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        012072  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        012076  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        012102  012737  000004  001226          MOV     #4,$TESTN       ;:SET TEST NUMBER IN APT MAIL BOX
    323
    324                                 ;;*****************************************************************
    325                                 ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
    326
    327 012110                          1$:
    328
    329                                 ;PREPARE DEVICE FOR TEST
    330 012110  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        012114  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        012116  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
        012120  000240                          NOP                     ;RETURN HERE IF ERROR
        012122  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        012124  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
    331
    332                                 ;LOAD PARAMETERS AND GENERATE DATA BUFFER
    333 012130                          2$:
    334 012130  012737  001057  001446          MOV     #559.,RMDC0             ;CYLINDER = 559.
    335 012136  012737  001000  001420          MOV     #TA2,RMDA0              ;TRACK = 2, SECTOR = 0
    336 012144  012737  104254  001416          MOV     #BUFONE,RMBA0           ;BUS ADDRESS
    337 012152  012737  177376  001414          MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
    338 012160  012737  010000  001444          MOV     #FMT16,RMOF0            ;16 BIT FORMAT
    339 012166  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
    340
    341                                 ;VERIFY THAT SECTOR IS NOT BAD
        012174  004737  034762                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        012200  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        012202  104401  066006                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        012206  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        012210  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
    342 012214                          3$:
    343 012214  012737  067456  001174          MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
    344 012222  012737  000001  001176          MOV     #1,$TMP1                ;RANGE OF PATTERN
    345 012230  004737  037214                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
    346
    347 012234  012702  001555                  MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
    348 012240  112722  000034                  MOVB    #RMDC,(R2)+
    349 012244  112722  000006                  MOVB    #RMDA,(R2)+
    350 012250  112722  000004                  MOVB    #RMBA,(R2)+
    351 012254  112722  000002                  MOVB    #RMWC,(R2)+
    352 012260  112722  000032                  MOVB    #RMOF,(R2)+
    353 012264  112722  000000                  MOVB    #RMCS1,(R2)+
```

```
354 012270  112712  000200                    MOVB    #200,(R2)               ;TERMINATE TABLE
355 012274                              4$:
356 012274  004737  040360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    012300  000404                            BR      5$              ;GO TO 5$ IF NO ERROR
    012302  000240                            NOP                     ;RETURN HERE IF ERROR
    012304  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    012306  000137  013256                    JMP     28$             ;GO TO 28$ IF ERROR
357 012312                              5$:
358
359                                     ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    012312  004737  040024                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
360
361                                     ;WAIT FOR COMMAND TO COMPLETE
    012316  004737  040732                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
362
363                                     ;GO EGT REGISTER STATUS
364 012322  004737  040110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    012326  000404                            BR      6$              ;GO TO 6$ IF NO ERROR
    012330  000240                            NOP                     ;RETURN HERE IF ERROR
    012332  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
    012334  000137  013256                    JMP     28$             ;GO TO 28$ IF ERROR
365 012340                              6$:
366
367                                     ;VERIFY RESULTS OF WRITE COMMAND
368 012340  004737  054166                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
    012344  000405                            BR      7$              ;GO TO 7$ IF NO ERROR
    012346  000240                            NOP                     ;RETURN HERE IF ERROR
    012350  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
    012352  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    012354  000137  013256                    JMP     28$             ;GO TO 28$ IF ERROR
369 012360                              7$:
370
371                                     ;MOVE LOOP ADDRESSES TO NEXT OPERATION
372 012360  012737  012370  001124            MOV     #8$,$LPERR
373 012366  000410                            BR      9$                      ;SKIP TO WRITE OPERATION
374
375                                     ;;*****************************************************************
376                                     ;LOOP #2        WRITE,WRITE CHECK DATA
377
378 012370                              8$:
379
380                                     ;PREPARE DEVICE FOR TEST
381 012370  004737  034032                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    012374  154130                            .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                       ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                       ;CLEAR CONTROLLER & SELECT DEVICE
                                                                       ;VERIFY CONTROLLER CLEAR OPERATION
                                                                       ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                       ;VERIFY PACK ACKNOWLEDGE
                                                                       ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                       ;VERIFY RECALIBRATION
    012376  000404                            BR      9$              ;GO TO 9$ IF NO ERROR
    012400  000240                            NOP                     ;RETURN HERE IF ERROR
    012402  104000                            EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    012404  000137  013256                    JMP     28$             ;GO TO 28$ IF ERROR
382 012410                              9$:
383
```

```
                                        ;SETUP PARAMETERS AND EXCUTE COMMAND
012410                                  10$:
012410  012737  177400  001414                MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
012416  012737  104260  001416                MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
012424  012737  000061  001412                MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
012432  012702  001555                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
012436  112722  000006                        MOVB    #RMDA,(R2)+
012442  112722  000034                        MOVB    #RMDC,(R2)+
012446  112722  000032                        MOVB    #RMOF,(R2)+
012452  112722  000004                        MOVB    #RMBA,(R2)+
012456  112722  000002                        MOVB    #RMWC,(R2)+
012462  112722  000000                        MOVB    #RMCS1,(R2)+
012466  112722  000200                        MOVB    #200,(R2)+              ;TERMINATE TABLE

012472  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
012476  000404                              BR      11$             ;GO TO 11$ IF NO ERROR
012500  000240                              NOP                     ;RETURN HERE IF ERROR
012502  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
012504  000137  013256                      JMP     28$             ;GO TO 28$ IF ERROR
012510                                  11$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
012510  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
012514  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE COMMAND
012520  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012524  000404                              BR      12$             ;GO TO 12$ IF NO ERROR
012526  000240                              NOP                     ;RETURN HERE IF ERROR
012530  104000                              EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
012532  000137  013256                      JMP     28$             ;GO TO 28$ IF ERROR
012536                                  12$:

                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
012536  004737  041126                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
012542  000405                              BR      13$             ;GO TO 13$ IF NO ERROR
012544  000240                              NOP                     ;RETURN HERE IF ERROR
012546  104000                              EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
012550  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012552  000137  013256                      JMP     28$             ;GO TO 28$ IF ERROR
012556                                  13$:
012556  004737  054166                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
012562  000405                              BR      14$             ;GO TO 14$ IF NO ERROR
012564  000240                              NOP                     ;RETURN HERE IF ERROR
012566  104000                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
012570  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012572  000137  013256                      JMP     28$             ;GO TO 28$ IF ERROR
012576                                  14$:
012576  004737  041760                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
012602  000405                              BR      15$             ;GO TO 15$ IF NO ERROR
012604  000240                              NOP                     ;RETURN HERE IF ERROR
012606  104000                              EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
012610  004736                              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
012612  000137  013256                      JMP     28$             ;GO TO 28$ IF ERROR
012616                                  15$:
```

```
                                        ;CHANGE LOOP ADDRESSES
012616  012737  012630  001124                MOV     #16$,$LPERR

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

012624  012703  000001                        MOV     #1,R3                   ;R3+WCE BIT POSITION
012630                                  16$:
012630  050337  105256                        BIS     R3,BUFTWO-2             ;CHANGE LAST WORD OF BUFFER

                                        ;PREPARE DEVICE FOR TEST
012634  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
012640  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
012642  000404                                BR      17$             ;GO TO 17$ IF NO ERROR
012644  000240                                NOP                     ;RETURN HERE IF ERROR
012646  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
012650  000137  013256                        JMP     28$             ;GO TO 28$ IF ERROR
012654                                  17$:

                                        ;READ DATA FROM DEVICE
012654                                  18$:
012654  012737  000051  001412                MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
012662  012702  001555                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
012666  112722  000006                        MOVB    #RMDA,(R2)+
012672  112722  000032                        MOVB    #RMOF,(R2)+
012676  112722  000034                        MOVB    #RMDC,(R2)+
012702  112722  000004                        MOVB    #RMBA,(R2)+
012706  112722  000002                        MOVB    #RMWC,(R2)+
012712  112722  000000                        MOVB    #RMCS1,(R2)+
012716  112712  000200                        MOVB    #200,(R2)

012722  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
012726  000404                                BR      19$             ;GO TO 19$ IF NO ERROR
012730  000240                                NOP                     ;RETURN HERE IF ERROR
012732  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
012734  000137  013256                        JMP     28$             ;GO TO 28$ IF ERROR
012740                                  19$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
012740  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
012744  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
012750  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012754  000404                                BR      20$             ;GO TO 20$ IF NO ERROR
012756  000240                                NOP                     ;RETURN HERE IF ERROR
012760  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
```

```
                012762  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
    452 012766                                  20$:
    453
    454                                         ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
    455 012766  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        012772  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
        012774  000240                          NOP                     ;RETURN HERE IF ERROR
        012776  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        013000  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013002  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
    456 013006                                  21$:
    457 013006  032737  040000  001346          BIT     #WCE,RMCS2I             ;WAS "WCE" DETECTED??
    458 013014  001030                          BNE     23$                     ;YES!!
    459
    460 013016  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        013022  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
        013024  000240                          NOP                     ;RETURN HERE IF ERROR
        013026  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        013030  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013032  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
    461 013036                                  22$:
    462 013036  013737  001346  001140          MOV     RMCS2I,$GDDAT           ;EXPECTED STATUS
    463 013044  052737  040000  001140          BIS     #WCE,$GDDAT
    464 013052  013737  001346  001142          MOV     RMCS2I,$BDDAT           ;RECEIVED STATUS
    465 013060  010337  001174                  MOV     R3,$TMP0                ;FAILING BIT POSITION
    466 013064  012737  105256  001176          MOV     #BUFTWO-2,$TMP1         ;FAILING ADDRESS
    467 013072  104337                          EMT     337
    468 013074  000470                          BR      28$
    469 013076                                  23$:
    470 013076  112737  000022  001526          MOVB    #RMDB,GETINX    ;SETUP GET INDEX TABLE
        013104  112737  000200  001527          MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
        013112  004737  040110                  JSR     PC,GET          ;GO READ RMDB VIA GET SUBROUTINE
        013116  000404                          BR      24$             ;GO TO 24$ IF NO ERROR
        013120  000240                          NOP                     ;RETURN HERE IF ERROR
        013122  104000                          EMT                     ;ERROR DEFINED BY GET SUBROUTINE
        013124  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
    471 013130                                  24$:
    472 013130  013737  001360  001142          MOV     RMDBI,$BDDAT            ;RECEIVED DATA
    473 013136  013737  105256  001140          MOV     BUFTWO-2,$GDDAT         ;EXPECTED DATA
    474 013144  040337  001140                  BIC     R3,$GDDAT
    475 013150  012737  105256  001134          MOV     #BUFTWO-2,$GDADR        ;EXPECTED ADDRESS
    476 013156  013737  001342  001136          MOV     RMBAI,$BDADR            ;RECEIVED ADDRESS
    477 013164  162737  000002  001136          SUB     #2,$BDADR               ;ADJUST BUS ADDRESS
    478 013172  023737  001134  001136          CMP     $GDADR,$BDADR           ;ADDRESSES OK??
    479 013200  001402                          BEQ     25$                     ;YES!!
    480 013202  104340                          EMT     340
    481 013204  000424                          BR      28$
    482 013206  023737  001140  001142  25$:    CMP     $GDDAT,$BDDAT           ;DATA OK??
    483 013214  001402                          BEQ     26$                     ;YES!!
    484 013216  104341                          EMT     341
    485 013220  000416                          BR      28$
    486 013222                                  26$:
    487
    488 013222  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        013226  000405                          BR      27$             ;GO TO 27$ IF NO ERROR
        013230  000240                          NOP                     ;RETURN HERE IF ERROR
        013232  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
```

```
013234  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
013236  000137  013256                  JMP     28$             ;GO TO 28$ IF ERROR
489 013242                      27$:
490 013242  040337  105256              BIC     R3,BUFTWO-2             ;RESTORE DATA PATTERN
491 013246  006303                      ASL     R3              ;SHIFT TO NEXT BIT
492 013250  001402                      BEQ     28$             ;EXIT IF DONE
493 013252  000137  012630              JMP     16$             ;REPEAT TEST FOR NEXT DATA BIT
494 013256                      28$:
495
496                             ;;*****************************************************************
                                ;*TEST 5         WRITE, READ ONES
                                ;;*****************************************************************
013256                          TST5:
013256  000004                          SCOPE                   ;SCOPE CALL
013260  000240                          NOP                     ;START OF TEST
013262  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
013266  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
013272  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
013276  012737  000005  001226          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
497
498                             ;;*****************************************************************
                                ;LOOP #1        FORMAT,WRITE,READ
500
501 013304                      1$:
502
503                             ;PREPARE THE DEVICE FOR TEST
504 013304  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
013310  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
013312  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
013314  000240                          NOP                     ;RETURN HERE IF ERROR
013316  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
013320  000137  014264                  JMP     25$             ;GO TO 25$ IF ERROR
505
506                             ;LOAD PARAMETERS AND GENERATE DATA BUFFER
507 013324                      2$:
508 013324  012737  001057  001446      MOV     #559.,RMDC0             ;CYLINDER = 559.
509 013332  012737  001000  001420      MOV     #TA2,RMDA0              ;TRACK = 2, SECTOR = 0
510 013340  012737  104254  001416      MOV     #BUFONE,RMBA0           ;BUS ADDRESS
511 013346  012737  177376  001414      MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
512 013354  012737  010000  001444      MOV     #FMT16,RMOF0            ;16 BIT FORMAT
513 013362  012737  000063  001412      MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
514
515                             ;VERIFY THAT SECTOR IS NOT BAD
013370  004737  034762                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
013374  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
013376  104401  066006                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
013402  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
013404  000137  014264                  JMP     25$             ;GO TO 25$ IF ERROR
516 013410                      3$:
517 013410  012737  067414  001174      MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
```

T5 WRITE, READ ONES

```
                                        MOV     #1,$TMP1                ;RANGE OF PATTERN
                                        JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
                                        MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
                                        MOVB    #RMDC,(R2)+
                                        MOVB    #RMDA,(R2)+
                                        MOVB    #RMBA,(R2)+
                                        MOVB    #RMWC,(R2)+
                                        MOVB    #RMOF,(R2)+
                                        MOVB    #RMCS1,(R2)+
                                        MOVB    #200,(R2)               ;TERMINATE TABLE
                                4$:

                                ;FORMAT THE DRIVE
                                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                                        BR      5$              ;GO TO 5$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                                        JMP     25$             ;GO TO 25$ IF ERROR
                                5$:

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

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

                                ;GO READ STATUS FOR FORMAT OPERATION
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      6$              ;GO TO 6$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     25$             ;GO TO 25$ IF ERROR
                                6$:

                                ;VERIFY NO ERRORS DURING FORMAT
                                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                                        BR      7$              ;GO TO 7$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                                        JMP     25$             ;GO TO 25$ IF ERROR
                                7$:

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

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

                                8$:


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

```
        013570  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
        013572  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
        013574  000240                          NOP                     ;RETURN HERE IF ERROR
        013576  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        013600  000137  014264                  JMP     25$             ;GO TO 25$ IF ERROR
    560 013604                          9$:
    561
    562                                 ;WRITE DATA TO THE DRIVE
    563 013604                          10$:
    564 013604  012737  104260  001416        MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
    565 013612  012737  177400  001414        MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
    566 013620  012737  000061  001412        MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
    567 013626  012702  001555                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    568 013632  112722  000006                MOVB    #RMDA,(R2)+
    569 013636  112722  000034                MOVB    #RMDC,(R2)+
    570 013642  112722  000032                MOVB    #RMOF,(R2)+
    571 013646  112722  000004                MOVB    #RMBA,(R2)+
    572 013652  112722  000002                MOVB    #RMWC,(R2)+
    573 013656  112722  000000                MOVB    #RMCS1,(R2)+
    574 013662  112722  000200                MOVB    #200,(R2)+              ;TERMINATE TABLE
    575
    576 013666  004737  040360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        013672  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
        013674  000240                          NOP                     ;RETURN HERE IF ERROR
        013676  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        013700  000137  014264                  JMP     25$             ;GO TO 25$ IF ERROR
    577 013704                          11$:
    578
    579                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        013704  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    580
    581                                 ;WAIT FOR COMMAND TO COMPLETE
        013710  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    582
    583                                 ;GO READ STATUS FOR WRITE COMMAND
    584 013714  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        013720  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
        013722  000240                          NOP                     ;RETURN HERE IF ERROR
        013724  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        013726  000137  014264                  JMP     25$             ;GO TO 25$ IF ERROR
    585 013732                          12$:
    586
    587                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
    588 013732  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        013736  000405                          BR      13$             ;GO TO 13$ IF NO ERROR
        013740  000240                          NOP                     ;RETURN HERE IF ERROR
        013742  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        013744  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        013746  000137  014264                  JMP     25$             ;GO TO 25$ IF ERROR
    589 013752                          13$:
```

T5 WRITE, READ ONES

```
590 013752  004737  054166              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
    013756  000405                      BR      14$             ;GO TO 14$ IF NO ERROR
    013760  000240                      NOP                     ;RETURN HERE IF ERROR
    013762  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
    013764  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    013766  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
591 013772                          14$:
592 013772  004737  041760              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
    013776  000405                      BR      15$             ;GO TO 15$ IF NO ERROR
    014000  000240                      NOP                     ;RETURN HERE IF ERROR
    014002  104000                      EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
    014004  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    014006  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
593 014012                          15$:
594
595                                 ;CHANGE LOOP ADDRESSES
596 014012  012737  014022  001124      MOV     #16$,$LPERR
597 014020  000410                      BR      17$                     ;SKIP TO NEXT OPERATION
598
599                                 ;;*****************************************************************
600                                 ;LOOP #3        READ
601
602 014022                          16$:
603
604                                 ;PREPARE DEVICE FOR TEST
605 014022  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    014026  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
    014030  000404                      BR      17$             ;GO TO 17$ IF NO ERROR
    014032  000240                      NOP                     ;RETURN HERE IF ERROR
    014034  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    014036  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
606 014042                          17$:
607
608                                 ;READ DATA FROM DEVICE
609 014042                          18$:
610 014042  012737  105264  001416      MOV     #BUFTWO+4,RMBAO         ;CHANGE MEMORY ADDRESS
611 014050  012737  000071  001412      MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
612 014056  012702  001555              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
613 014062  112722  000006              MOVB    #RMDA,(R2)+
614 014066  112722  000032              MOVB    #RMOF,(R2)+
615 014072  112722  000034              MOVB    #RMDC,(R2)+
616 014076  112722  000004              MOVB    #RMBA,(R2)+
617 014102  112722  000002              MOVB    #RMWC,(R2)+
618 014106  112722  000000              MOVB    #RMCS1,(R2)+
619 014112  112712  000200              MOVB    #200,(R2)
620
621 014116  004737  040360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    014122  000404                      BR      19$             ;GO TO 19$ IF NO ERROR
    014124  000240                      NOP                     ;RETURN HERE IF ERROR
    014126  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
```

T5 WRITE, READ ONES

```
014130  000137  014264                    JMP     25$             ;GO TO 25$ IF ERROR
622 014134                          19$:
623
624                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
014134  004737  040024                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
625
626                                 ;WAIT FOR COMMAND TO COMPLETE
014140  004737  040732                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
627
628                                 ;GO READ STATUS FOR READ OPERATION
629 014144  004737  040110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
014150  000404                      BR      20$             ;GO TO 20$ IF NO ERROR
014152  000240                      NOP                     ;RETURN HERE IF ERROR
014154  104000                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
014156  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
630 014162                          20$:
631
632                                 ;CHECK FOR ERRORS DURING READ OPERATION
633 014162  004737  041126                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
014166  000405                      BR      21$             ;GO TO 21$ IF NO ERROR
014170  000240                      NOP                     ;RETURN HERE IF ERROR
014172  104000                      EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
014174  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014176  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
634 014202                          21$:
635 014202  004737  054166                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
014206  000405                      BR      22$             ;GO TO 22$ IF NO ERROR
014210  000240                      NOP                     ;RETURN HERE IF ERROR
014212  104000                      EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
014214  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014216  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
636 014222                          22$:
637 014222  004737  041760                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
014226  000405                      BR      23$             ;GO TO 23$ IF NO ERROR
014230  000240                      NOP                     ;RETURN HERE IF ERROR
014232  104000                      EMT                     ;ERROR # DEFINED BY SECEPR SUBROUTINE
014234  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014236  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
638 014242                          23$:
639 014242  004737  037462                JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
014246  104260                      .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
014250  105264                      .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
014252  000404                      BR      24$             ;GO TO 24$ IF NO ERROR
014254  000240                      NOP                     ;RETURN HERE IF ERROR
014256  104000                      EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
014260  000137  014264              JMP     25$             ;GO TO 25$ IF ERROR
640 014264                          24$:
641
642 014264                          25$:
643
644
                                    ;;*********************************************************************
                                    ;*TEST 6        WRITE, WRITE CHECK ONES
                                    ;;*********************************************************************
                                    TST6:
014264
014264  000004                      SCOPE                   ;SCOPE CALL
014266  000240                      NOP                     ;START OF TEST
014270  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
```

T6 WRITE, WRITE CHECK ONES

```
014274  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
014300  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
014304  012737  000006  001226          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
645
646                             ;;*****************************************************************
647                             ;LOOP #1        FORMAT,WRITE,WRITE CHECK DATA
648
649 014312                      1$:
650
651                             ;PREPARE THE DEVICE FOR TEST
652 014312  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    014316  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
    014320  000404                      BR      2$              ;GO TO 2$ IF NO ERROR
    014322  000240                      NOP                     ;RETURN HERE IF ERROR
    014324  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    014326  000137  015242              JMP     24$             ;GO TO 24$ IF ERROR
653
654                             ;LOAD PARAMETERS AND GENERATE DATA BUFFER
655 014332                      2$:
656 014332  012737  001057  001446      MOV     #559.,RMDC0             ;CYLINDER = 559.
657 014340  012737  001000  001420      MOV     #TA2,RMDA0              ;TRACK = 2, SECTOR = 0
658 014346  012737  104254  001416      MOV     #BUFONE,RMBA0           ;BUS ADDRESS
659 014354  012737  177376  001414      MOV     #-258.,RMWC0            ;2 + 256. WORDS (2'S COMP)
660 014362  012737  010000  001444      MOV     #FMT16,RMOF0            ;16 BIT FORMAT
661 014370  012737  000063  001412      MOV     #WH!GO,RMCS10           ;WRITE HEADER AND DATA COMMAND
662
663                             ;VERIFY THAT SECTOR IS NOT BAD
    014376  004737  034762              JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
    014402  000405                      BR      3$              ;GO TO 3$ IF NO ERROR
    014404  104401  066006              TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    014410  104000                      EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    014412  000137  015242              JMP     24$             ;GO TO 24$ IF ERROR
664 014416                      3$:
665 014416  012737  067414  001174      MOV     #ONES,$TMP0             ;STARTING ADDRESS OF PATTERN
666 014424  012737  000001  001176      MOV     #1,$TMP1                ;RANGE OF PATTERN
667 014432  004737  037214              JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
668
669                             ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
670 014436  012702  001555              MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
671 014442  112722  000034              MOVB    #RMDC,(R2)+
672 014446  112722  000006              MOVB    #RMDA,(R2)+
673 014452  112722  000004              MOVB    #RMBA,(R2)+
674 014456  112722  000002              MOVB    #RMWC,(R2)+
675 014462  112722  000032              MOVB    #RMOF,(R2)+
676 014466  112722  000000              MOVB    #RMCS1,(R2)+
677 014472  112712  000200              MOVB    #200,(R2)               ;TERMINATE TABLE
678 014476                      4$:
679
680                             ;FORMAT THE DRIVE
681 014476  004737  040360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
```

```
        014502  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
        014504  000240                          NOP                     ;RETURN HERE IF ERROR
        014506  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        014510  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
   682  014514                          5$:
   683
   684                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        014514  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
   685
   686                                  ;WAIT FOR COMMAND TO COMPLETE
        014520  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
   687
   688                                  ;GO READ STATUS FOR FORMAT OPERATION
   689  014524  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        014530  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        014532  000240                          NOP                     ;RETURN HERE IF ERROR
        014534  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014536  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
   690  014542                          6$:
   691
   692                                  ;VERIFY NO ERRORS DURING FORMAT
   693  014542  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        014546  000405                          BR      7$              ;GO TO 7$ IF NO ERROR
        014550  000240                          NOP                     ;RETURN HERE IF ERROR
        014552  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        014554  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014556  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
   694  014562                          7$:
   695
   696                                  ;MOVE LOOP ADDRESSES TO NEXT OPERATION
   697  014562  012737  014572  001124          MOV     #8$,$LPERR
   698  014570  000410                          BR      9$                      ;SKIP TO WRITE OPERATION
   699
   700                                  ;;*****************************************************************
   701                                  ;LOOP #2        WRITE,WRITE CHECK DATA
   702
   703  014572                          8$:
   704
   705
   706                                  ;PREPARE DEVICE FOR WRITE OPERATION
   707  014572  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        014576  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        014600  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
        014602  000240                          NOP                     ;RETURN HERE IF ERROR
        014604  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        014606  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
   708  014612                          9$:
   709
   710                                  ;WRITE DATA TO THE DRIVE
   711  014612                          10$:
```

```
    712 014612  012737  104260  001416          MOV     #SUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
    713 014620  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
    714 014626  012737  000061  001412          MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
    715 014634  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
    716 014640  112722  000006                  MOVB    #RMDA,(R2)+
    717 014644  112722  000034                  MOVB    #RMDC,(R2)+
    718 014650  112722  000032                  MOVB    #RMOF,(R2)+
    719 014654  112722  000004                  MOVB    #RMBA,(R2)+
    720 014660  112722  000002                  MOVB    #RMWC,(R2)+
    721 014664  112722  000000                  MOVB    #RMCS1,(R2)+
    722 014670  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
    723
    724 014674  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        014700  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
        014702  000240                          NOP                     ;RETURN HERE IF ERROR
        014704  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        014706  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
    725 014712                          11$:
    726
    727                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        014712  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
    728
    729                                 ;WAIT FOR COMMAND TO COMPLETE
        014716  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
    730
    731                                 ;GO READ STATUS FOR WRITE COMMAND
    732 014722  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        014726  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
        014730  000240                          NOP                     ;RETURN HERE IF ERROR
        014732  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014734  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
    733 014740                          12$:
    734
    735                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
    736 014740  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        014744  000405                          BR      13$             ;GO TO 13$ IF NO ERROR
        014746  000240                          NOP                     ;RETURN HERE IF ERROR
        014750  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        014752  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014754  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
    737 014760                          13$:
    738 014760  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        014764  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
        014766  000240                          NOP                     ;RETURN HERE IF ERROR
        014770  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        014772  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014774  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
    739 015000                          14$:
    740 015000  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        015004  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
        015006  000240                          NOP                     ;RETURN HERE IF ERROR
        015010  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        015012  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015014  000137  015242                  JMP     24$             ;GO TO 24$ IF ERROR
    741 015020                          15$:
    742
    743                                 ;CHANGE LOOP ADDRESSES
```

```
744 015020  012737  015030  001124          MOV     #16$,$LPERR
745 015026  000410                          BR      17$                     ;SKIP TO NEXT OPERATION
746
747                                 ;;*****************************************************************
748                                 ;LOOP #3        WRITE CHECK DATA
749
750 015030                          16$:
751
752                                 ;PREPARE DEVICE FOR WRITE CHECK DATA OPERATION
753 015030  004737  034032                  JSR     PC,TSTPRP               ;PREPARE DEVICE FOR TEST
    015034  154130                          .WORD   154130                  ;TASK DESCRIPTOR AS FOLLOWS:
                                                                            ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                            ;VERIFY PACK ACKNOWLEDGE
                                                                            ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                            ;VERIFY RECALIBRATION
    015036  000404                          BR      17$                     ;GO TO 17$ IF NO ERROR
    015040  000240                          NOP                             ;RETURN HERE IF ERROR
    015042  104000                          EMT                             ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    015044  000137  015242                  JMP     24$                     ;GO TO 24$ IF ERROR
754 015050                          17$:
755
756                                 ;WRITE CHECK DATA DATA FROM DEVICE
757 015050                          18$:
758 015050  012737  000051  001412          MOV     #WCD!GO,RMCS10          ;WRITE CHECK DATA DATA COMMAND
759 015056  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
760 015062  112722  000006                  MOVB    #RMDA,(R2)+
761 015066  112722  000032                  MOVB    #RMOF,(R2)+
762 015072  112722  000034                  MOVB    #RMDC,(R2)+
763 015076  112722  000004                  MOVB    #RMBA,(R2)+
764 015102  112722  000002                  MOVB    #RMWC,(R2)+
765 015106  112722  000000                  MOVB    #RMCS1,(R2)+
766 015112  112712  000200                  MOVB    #200,(R2)
767
768 015116  004737  040360                  JSR     PC,PUT                  ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    015122  000404                          BR      19$                     ;GO TO 19$ IF NO ERROR
    015124  000240                          NOP                             ;RETURN HERE IF ERROR
    015126  104000                          EMT                             ;ERROR # DEFINED BY PUT SUBROUTINE
    015130  000137  015242                  JMP     24$                     ;GO TO 24$ IF ERROR
769 015134                          19$:
770
771                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    015134  004737  040024                  JSR     PC,GETSTS               ;GO TO GETSTS SUBROUTINE
772
773                                 ;WAIT FOR COMMAND TO COMPLETE
    015140  004737  040732                  JSR     PC,TIMOUT               ;GO TO TIMOUT SUBROUTINE
774
775                                 ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
776 015144  004737  040110                  JSR     PC,GET                  ;GO READ REGISTER(S) WITH GET SUBROUTINE
    015150  000404                          BR      20$                     ;GO TO 20$ IF NO ERROR
    015152  000240                          NOP                             ;RETURN HERE IF ERROR
    015154  104000                          EMT                             ;ERROR # DEFINED BY GET SUBROUTINE
    015156  000137  015242                  JMP     24$                     ;GO TO 24$ IF ERROR
777 015162                          20$:
778
```

```
                                        ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
015162  004737  041126                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
015166  000405                                BR      21$             ;GO TO 21$ IF NO ERROR
015170  000240                                NOP                     ;RETURN HERE IF ERROR
015172  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
015174  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015176  000137  015242                        JMP     24$             ;GO TO 24$ IF ERROR
015202                                  21$:
015202  004737  054166                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
015206  000405                                BR      22$             ;GO TO 22$ IF NO ERROR
015210  000240                                NOP                     ;RETURN HERE IF ERROR
015212  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
015214  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015216  000137  015242                        JMP     24$             ;GO TO 24$ IF ERROR
015222                                  22$:
015222  004737  041760                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
015226  000405                                BR      23$             ;GO TO 23$ IF NO ERROR
015230  000240                                NOP                     ;RETURN HERE IF ERROR
015232  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
015234  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015236  000137  015242                        JMP     24$             ;GO TO 24$ IF ERROR
015242                                  23$:

015242                                  24$:


                                        ;;*************************************************************
                                        ;*TEST 7        WRITE, WRITE CHECK ONES W/ WCE ERROR
                                        ;;*************************************************************
015242                                  TST7:
015242  000004                                SCOPE                   ;SCOPE CALL
015244  000240                                NOP                     ;START OF TEST
015246  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
015252  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
015256  013701  001466                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
015262  012737  000007  001226                MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX


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

015270                                  1$:

                                        ;PREPARE THE DEVICE FOR FORMAT OPERATION
015270  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
015274  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
015276  000404                                BR      2$              ;GO TO 2$ IF NO ERROR
015300  000240                                NOP                     ;RETURN HERE IF ERROR
015302  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
015304  000137  016440                        JMP     28$             ;GO TO 28$ IF ERROR
015310                                  2$:
```

```
800                                     ;LOAD PARAMETERS AND GENERATE DATA BUFFER
801 015310  012737  001057  001446        MOV     #559.,RMDCO             ;CYLINDER = 559.
802 015316  012737  001000  001420        MOV     #TA2,RMDAO              ;TRACK = 2, SECTOR = 0
803 015324  012737  104254  001416        MOV     #BUFONE,RMBAC           ;BUS ADDRESS
804 015332  012737  177376  001414        MOV     #-258.,RMWCO            ;WORD COUNT = 1 SECTOR
805 015340  012737  010000  001444        MOV     #FMT16,RMOFO            ;16 BIT FORMAT
806 015346  012737  000063  001412        MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
807
808                                     ;VERIFY THAT SECTOR IS NOT BAD
    015354  004737  034762                MOV     PC,BADSCT       ;CALL BAD SECTOR MODULE
    015360  000405                        BR      3$              ;GO TO 3$ IF NO ERROR
    015362  104401  066006                TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
    015366  104000                        EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
    015370  000137  016440                JMP     28$             ;GO TO 28$ IF ERROR
809 015374                              3$:
810 015374  012737  067414  001174        MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
811 015402  012737  000001  001176        MOV     #1,$TMP1                ;RANGE OF PATTERN
812 015410  004737  037214                JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
813
814                                     ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
815 015414  012702  001555                MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
816 015420  112722  000034                MOVB    #RMDC,(R2)+
817 015424  112722  000006                MOVB    #RMDA,(R2)+
818 015430  112722  000004                MOVB    #RMBA,(R2)+
819 015434  112722  000002                MOVB    #RMWC,(R2)+
820 015440  112722  000032                MOVB    #RMOF,(R2)+
821 015444  112722  000000                MOVB    #RMCS1,(R2)+
822 015450  112712  000200                MOVB    #200,(R2)               ;TERMINATE TABLE
823 015454                              4$:
824
825                                     ;FORMAT THE DRIVE
826 015454  004737  040360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    015460  000404                        BR      5$              ;GO TO 5$ IF NO ERROR
    015462  000240                        NOP                     ;RETURN HERE IF ERROR
    015464  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    015466  000137  016440                JMP     28$             ;GO TO 28$ IF ERROR
827 015472                              5$:
828
829                                     ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    015472  004737  040024                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
830
831                                     ;WAIT FOR COMMAND TO COMPLETE
    015476  004737  040732                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
832
833                                     ;GO READ STATUS FOR FORMAT OPERATION
834 015502  004737  040110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    015506  000404                        BR      6$              ;GO TO 6$ IF NO ERROR
    015510  000240                        NOP                     ;RETURN HERE IF ERROR
    015512  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
    015514  000137  016440                JMP     28$             ;GO TO 28$ IF ERROR
835 015520                              6$:
836
837                                     ;VERIFY NO ERRORS DURING FORMAT
838 015520  004737  054166                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
    015524  000405                        BR      7$              ;GO TO 7$ IF NO ERROR
    015526  000240                        NOP                     ;RETURN HERE IF ERROR
    015530  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
```

```
015532  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015534  000137  016440                   JMP     28$             ;GO TO 28$ IF ERROR
839 015540                      7$:
840
841                             ;MOVE LOOP ADDRESSES TO NEXT OPERATION
842 015540  012737  015550  001124      MOV     #8$,$LPERR
843 015546  000410                      BR      9$                      ;SKIP TO WRITE OPERATION
844
845                             ;;*****************************************************************
846                             ;LOOP #2        WRITE,WRITE CHECK DATA
847
848 015550                      8$:
849
850
851                             ;PREPARE DEVICE FOR WRITE OPERATION
852 015550  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    015554  154130                      .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
    015556  000404                      BR      9$              ;GO TO 9$ IF NO ERROR
    015560  000240                      NOP                     ;RETURN HERE IF ERROR
    015562  104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    015564  000137  016440              JMP     28$             ;GO TO 28$ IF ERROR
853 015570                      9$:
854
855                             ;WRITE DATA TO THE DRIVE
856 015570                      10$:
857 015570  012737  177400  001414      MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
858 015576  012737  104260  001416      MOV     #BUFONE+4,RMBAO         ;CHANGE MEMORY ADDRESS
859 015604  012737  000061  001412      MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
860 015612  012702  001555              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
861 015616  112722  000006              MOVB    #RMDA,(R2)+
862 015622  112722  000034              MOVB    #RMDC,(R2)+
863 015626  112722  000032              MOVB    #RMOF,(R2)+
864 015632  112722  000004              MOVB    #RMBA,(R2)+
865 015636  112722  000002              MOVB    #RMWC,(R2)+
866 015642  112722  000000              MOVB    #RMCS1,(R2)+
867 015646  112722  000200              MOVB    #200,(R2)+              ;TERMINATE TABLE
868
869 015652  004737  04C360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    015656  000404                      BR      11$             ;GO TO 11$ IF NO ERROR
    015660  000240                      NOP                     ;RETURN HERE IF ERROR
    015662  104000                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    015664  000137  016440              JMP     28$             ;GO TO 28$ IF ERROR
870 015670                      11$:
871
872                             ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
    015670  004737  040024              JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
873
874                             ;WAIT FOR COMMAND TO COMPLETE
    015674  004737  040732              JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
875
```

```
                                        ;GO READ STATUS FOR WRITE COMMAND
015700  004737  040110                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
015704  000404                                  BR      12$             ;GO TO 12$ IF NO ERROR
015706  000240                                  NOP                     ;RETURN HERE IF ERROR
015710  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
015712  000137  016440                          JMP     28$             ;GO TO 28$ IF ERROR
015716                                  12$:

                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
015716  004737  041126                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
015722  000405                                  BR      13$             ;GO TO 13$ IF NO ERROR
015724  000240                                  NOP                     ;RETURN HERE IF ERROR
015726  104000                                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
015730  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015732  000137  016440                          JMP     28$             ;GO TO 28$ IF ERROR
015736                                  13$:
015736  004737  054166                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
015742  000405                                  BR      14$             ;GO TO 14$ IF NO ERROR
015744  000240                                  NOP                     ;RETURN HERE IF ERROR
015746  104000                                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
015750  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015752  000137  016440                          JMP     28$             ;GO TO 28$ IF ERROR
015756                                  14$:
015756  004737  041760                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
015762  000405                                  BR      15$             ;GO TO 15$ IF NO ERROR
015764  000240                                  NOP                     ;RETURN HERE IF ERROR
015766  104000                                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
015770  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
015772  000137  016440                          JMP     28$             ;GO TO 28$ IF ERROR
015776                                  15$:

                                        ;CHANGE LOOP ADDRESSES
015776  012737  016010  001124                  MOV     #16$,$LPERR

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

016004  012703  000001                          MOV     #1,R3                   ;R3=WCE BIT POSITION
016010                                  16$:
016010  040337  105256                          BIC     R3,BUFTWO-2             ;CHANGE LAST WORD OF BUFFER

                                        ;PREPARE DEVICE FOR WRITE CHECK DATA OPERATION
016014  004737  034032                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
016020  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
016022  000404                                  BR      17$             ;GO TO 17$ IF NO ERROR
016024  000240                                  NOP                     ;RETURN HERE IF ERROR
016026  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
016030  000137  016440                          JMP     28$             ;GO TO 28$ IF ERROR
016034                                  17$:
```

```
                                        ;READ DATA FROM DEVICE
016034                                  18$:
016034  012737  000051  001412                MOV     #WCD!GO,RMCS1O          ;WRITE CHECK DATA DATA COMMAND
016042  012702  001555                        MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
016046  112722  000006                        MOVB    #RMDA,(R2)+
016052  112722  000032                        MOVB    #RMOF,(R2)+
016056  112722  000034                        MOVB    #RMDC,(R2)+
016062  112722  000004                        MOVB    #RMBA,(R2)+
016066  112722  000002                        MOVB    #RMWC,(R2)+
016072  112722  000000                        MOVB    #RMCS1,(R2)+
016076  112712  000200                        MOVB    #200,(R2)

016102  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
016106  000404                                BR      19$             ;GO TO 19$ IF NO ERROR
016110  000240                                NOP                     ;RETURN HERE IF ERROR
016112  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
016114  000137  016440                        JMP     28$             ;GO TO 28$ IF ERROR
016120                                  19$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
016120  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
016124  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE CHECK DATA OPERATION
016130  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016134  000404                                BR      20$             ;GO TO 20$ IF NO ERROR
016136  000240                                NOP                     ;RETURN HERE IF ERROR
016140  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
016142  000137  016440                        JMP     28$             ;GO TO 28$ IF ERROR
016146                                  20$:

                                        ;CHECK FOR ERRORS DURING WRITE CHECK DATA OPERATION
016146  004737  041126                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
016152  000405                                BR      21$             ;GO TO 21$ IF NO ERROR
016154  000240                                NOP                     ;RETURN HERE IF ERROR
016156  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
016160  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016162  000137  016440                        JMP     28$             ;GO TO 28$ IF ERROR
016166                                  21$:
016166  032737  040000  001346                BIT     #WCE,RMCS2I             ;WAS 'WCE" DETECTED??
016174  001030                                BNE     23$                     ;YES!!
016176  004737  054166                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
016202  000405                                BR      22$             ;GO TO 22$ IF NO ERROR
016204  000240                                NOP                     ;RETURN HERE IF ERROR
016206  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
016210  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016212  000137  016440                        JMP     28$             ;GO TO 28$ IF ERROR
016216                                  22$:
016216  013737  001346  001140                MOV     RMCS2I,$GDDAT           ;EXPECTED STATUS
016224  052737  040000  001140                BIS     #WCE,$GDDAT
016232  013737  001346  001142                MOV     RMCS2I,$BDDAT           ;RECEIVED STATUS
016240  010337  001174                        MOV     R3,$TMP0                ;FAILING BIT POSITION
016244  012737  105256  001176                MOV     #BUFTWO-2,$TMP1         ;FAILING ADDRESS
016252  104337                                EMT     337
016254  000471                                BR      28$
```

```
939
940 016256                                  23$:
941 016256  112737  000022  001526          MOVB    #RMDB,GETINX    ;SETUP GET INDEX TABLE
    016264  112737  000200  001527          MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
    016272  012737  016440  001360          MOV     #28$,RMDBI      ;SET RMDB INPUT BUFFER = 28$
    016300  004737  040110                  JSR     PC,GET          ;GO READ RMDB VIA GET SUBROUTINE
    016304  000402                          BR      24$             ;GO TO 24$ IF NO ERROR
    016306  000240                          NOP                     ;RETURN HERE IF ERROR
    016310  104000                          EMT                     ;ERROR DEFINED BY GET SUBROUTINE
942
943 016312                                  24$:
944 016312  013737  001360  001142          MOV     RMDBI,$BDDAT            ;RECEIVED DATA
945 016320  013737  105256  001140          MOV     BUFTWO-2,$GDDAT         ;EXPECTED DATA
946 016326  050337  001140                  BIS     R3,$GDDAT
947 016332  012737  105256  001134          MOV     #BUFTWO-2,$GDADR        ;EXPECTED ADDRESS
948 016340  013737  001342  001136          MOV     RMBAI,$BDADR            ;RECEIVED ADDRESS
949 016346  162737  000002  001136          SUB     #2,$BDADR               ;CORRECT MEMORY ADDRESS
950 016354  023737  001134  001136          CMP     $GDADR,$BDADR           ;ADDRESSES OK??
951 016362  001402                          BEQ     25$                     ;YES!!
952 016364  104340                          EMT     340
953 016366  000424                          BR      28$
954 016370  023737  001140  001142  25$:    CMP     $GDDAT,$BDDAT           ;DATA OK??
955 016376  001402                          BEQ     26$                     ;YES!!
956 016400  104341                          EMT     341
957 016402  000416                          BR      28$
958 016404                                  26$:
959
960 016404  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
    016410  000405                          BR      27$             ;GO TO 27$ IF NO ERROR
    016412  000240                          NOP                     ;RETURN HERE IF ERROR
    016414  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
    016416  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    016420  000137  016440                  JMP     28$             ;GO TO 28$ IF ERROR
961 016424                                  27$:
962 016424  050337  105256                  BIS     R3,BUFTWO-2             ;RESTORE DATA PATTERN
963 016430  006303                          ASL     R3                      ;SHIFT TO NEXT BIT
964 016432  001402                          BEQ     28$                     ;EXIT IF DONE
965 016434  000137  016010                  JMP     16$                     ;REPEAT TEST FOR NEXT DATA BIT
966 016440                                  28$:
967
968                                         ;;*************************************************************************
                                            ;*TEST 10        WRITE, WRITE CHECK MULTIPLE SECTORS
                                            ;;*************************************************************************
    016440                                  TST10:
    016440  000004                          SCOPE                   ;SCOPE CALL
    016442  000240                          NOP                     ;START OF TEST
    016444  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
    016450  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    016454  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
    016460  012737  000010  001226          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
969
970                                         ;;*************************************************************************
971                                         ;LOOP #1        FORMAT,WRITE,WRITE CHECK
972
973 016466                                  1$:
974 016466  012737  010000  001444          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
975
```

```
                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
016474                                  2$:
016474  012737  001057  001446                MOV     #559.,RMDCO             ;CYLINDER = 559.
016502  012737  001000  001420                MOV     #TA2,RMDAO              ;TRACK = 2, SECTOR = 0
016510  012737  104254  001416                MOV     #BUFONE,RMBAO           ;BUS ADDRESS
016516  012737  176774  001414                MOV     #-258.*2,RMWCO          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
016524  012737  000063  001412                MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND

                                        ;PREPARE THE DEVICE FOR FORMAT OPERATION
016532  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
016536  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VEPIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
016540  000404                                BR      3$              ;GO TO 3$ IF NO ERROR
016542  000240                                NOP                     ;RETURN HERE IF ERROR
016544  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
016546  000137  017466                        JMP     25$             ;GO TO 25$ IF ERROR
016552                                  3$:

                                        ;VERIFY THAT SECTOR IS NOT BAD
016552  004737  034762                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
016556  000405                                BR      4$              ;GO TO 4$ IF NO ERROR
016560  104401  066006                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
016564  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
016566  000137  017466                        JMP     25$             ;GO TO 25$ IF ERROR
016572                                  4$:
016572  012737  067456  001174                MOV     #ZEROS,$TMP0            ;STARTING ADDRESS OF PATTERN
016600  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
016606  004737  037214                        JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
016612  012702  001555                        MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
016616  112722  000034                        MOVB    #RMDC,(R2)+
016622  112722  000006                        MOVB    #RMDA,(R2)+
016626  112722  000004                        MOVB    #RMBA,(R2)+
016632  112722  000002                        MOVB    #RMWC,(R2)+
016636  112722  000032                        MOVB    #RMOF,(R2)+
016642  112722  000000                        MOVB    #RMCS1,(R2)+
016646  112712  000200                        MOVB    #200,(R2)               ;TERMINATE TABLE
016652                                  5$:

                                        ;FORMAT THE DRIVE
016652  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
016656  000404                                BR      6$              ;GO TO 6$ IF NO ERROR
016660  000240                                NOP                     ;RETURN HERE IF ERROR
016662  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
016664  000137  017466                        JMP     25$             ;GO TO 25$ IF ERROR
016670                                  6$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
016670  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
```

```
                                        ;WAIT FOR COMMAND TO COMPLETE
016674  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR FORMAT OPERATION
016700  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016704  000404                                BR      7$              ;GO TO 7$ IF NO ERROR
016706  000240                                NOP                     ;RETURN HERE IF ERROR
016710  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
016712  000137  017466                        JMP     25$             ;GO TO 25$ IF ERROR
016716                                  7$:

                                        ;VERIFY NO ERRORS DURING FORMAT
016716  004737  054166                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
016722  000405                                BR      8$              ;GO TO 8$ IF NO ERROR
016724  000240                                NOP                     ;RETURN HERE IF ERROR
016726  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
016730  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016732  000137  017466                        JMP     25$             ;GO TO 25$ IF ERROR
016736                                  8$:

                                        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
016736  012737  017010  001124                MOV     #9$,$LPERR

                                        ;REGENERATE DATA BUFFER
016744  012737  177000  001414                MOV     #-256.*2,RMWCO  ;CHANGE WORD COUNT
016752  012737  104254  001416                MOV     #BUFONE,RMBAO           ;CHANGE BUS ADDRESS
016760  012737  000060  001412                MOV     #WD,RMCS1O              ;WRITE DATA
016766  012737  067414  001174                MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
016774  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
017002  004737  037214                        JSR     PC,GENBUF
017006  000410                                BR      10$                     ;SKIP TO WRITE OPERATION

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

017010                                  9$:


                                        ;PREPARE DEVICE FOR WRITE OPERATION
017010  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
017014  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
017016  000404                                BR      10$             ;GO TO 10$ IF NO ERROR
017020  000240                                NOP                     ;RETURN HERE IF ERROR
017022  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
017024  000137  017466                        JMP     25$             ;GO TO 25$ IF ERROR
017030                                  10$:

                                        ;WRITE DATA TO THE DRIVE
017030                                  11$:
017030  012737  000061  001412                MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
```

```
1046 017036  012702  001555                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1047 017042  112722  000006                MOVB    #RMDA,(R2)+
1048 017046  112722  000034                MOVB    #RMDC,(R2)+
1049 017052  112722  000032                MOVB    #RMOF,(R2)+
1050 017056  112722  000004                MOVB    #RMBA,(R2)+
1051 017062  112722  000002                MOVB    #RMWC,(R2)+
1052 017066  112722  000000                MOVB    #RMCS1,(R2)+
1053 017072  112722  000200                MOVB    #200,(R2)+              ;TERMINATE TABLE
1054
1055 017076  004737  040360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     017102  000404                        BR      12$             ;GO TO 12$ IF NO ERROR
     017104  000240                        NOP                     ;RETURN HERE IF ERROR
     017106  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     017110  000137  017466                JMP     25$             ;GO TO 25$ IF ERROR
1056 017114                        12$:
1057
1058                               ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     017114  004737  040024                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1059
1060                               ;WAIT FOR COMMAND TO COMPLETE
     017120  004737  040732                JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1061
1062
1063 017124  004737  040110                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     017130  000404                        BR      13$             ;GO TO 13$ IF NO ERROR
     017132  000240                        NOP                     ;RETURN HERE IF ERROR
     017134  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     017136  000137  017466                JMP     25$             ;GO TO 25$ IF ERROR
1064 017142                        13$:
1065
1066                               ;CHECK FOR ERRORS DURING WRITE OPERATION
1067 017142  004737  041126                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     017146  000405                        BR      14$             ;GO TO 14$ IF NO ERROR
     017150  000240                        NOP                     ;RETURN HERE IF ERROR
     017152  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     017154  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017156  000137  017466                JMP     25$             ;GO TO 25$ IF ERROR
1068 017162                        14$:
1069 017162  004737  054166                JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     017166  000405                        BR      15$             ;GO TO 15$ IF NO ERROR
     017170  000240                        NOP                     ;RETURN HERE IF ERROR
     017172  104000                        EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     017174  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017176  000137  017466                JMP     25$             ;GO TO 25$ IF ERROR
1070 017202                        15$:
1071 017202  004737  041760                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     017206  000405                        BR      16$             ;GO TO 16$ IF NO ERROR
     017210  000240                        NOP                     ;RETURN HERE IF ERROR
     017212  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     017214  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     017216  000137  017466                JMP     25$             ;GO TO 25$ IF ERROR
1072 017222                        16$:
1073
1074                               ;CHANGE LOOP ADDRESSES
1075 017222  012737  017232  001124        MOV     #17$,$LPERR
1076 017230  000410                        BR      18$                     ;SKIP TO NEXT OPERATION
1077
```

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

017232                          17$:

                                ;PREPARE DEVICE FOR READ OPERATION
017232  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
017236  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
017240  000404                          BR      18$             ;GO TO 18$ IF NO ERROR
017242  000240                          NOP                     ;RETURN HERE IF ERROR
017244  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
017246  000137  017466                  JMP     25$             ;GO TO 25$ IF ERROR
017252                          18$:

                                ;READ DATA FROM DEVICE
017252                          19$:
017252  012737  000051  001412          MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
017260  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
017264  112722  000006                  MOVB    #RMDA,(R2)+
017270  112722  000032                  MOVB    #RMOF,(R2)+
017274  112722  000034                  MOVB    #RMDC,(R2)+
017300  112722  000004                  MOVB    #RMBA,(R2)+
017304  112722  000002                  MOVB    #RMWC,(R2)+
017310  112722  000000                  MOVB    #RMCS1,(R2)+
017314  112712  000200                  MOVB    #200,(R2)

017320  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
017324  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
017326  000240                          NOP                     ;RETURN HERE IF ERROR
017330  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
017332  000137  017466                  JMP     25$             ;GO TO 25$ IF ERROR
017336                          20$:

                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
017336  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                ;WAIT FOR COMMAND TO COMPLETE
017342  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                ;GO READ STATUS FOR WRITE CHECK OPERATION
017346  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
017352  000404                          BR      21$             ;GO TO 21$ IF NO ERROR
017354  000240                          NOP                     ;RETURN HERE IF ERROR
017356  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
017360  000137  017466                  JMP     25$             ;GO TO 25$ IF ERROR
017364                          21$:

                                ;CHECK FOR ERRORS DURING WRITE CHECK OPERATION
017364  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
017370  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
```

T10 WRITE, WRITE CHECK MULTIPLE SECTORS

```
017372  000240                          NOP                     ;RETURN HERE IF ERROR
017374  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
017376  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
017400  000137  017466                  JMP     25$             ;GO TO 25$ IF ERROR
1112 017404                     22$:
1113 017404  004737  054166            JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
017410  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
017412  000240                          NOP                     ;RETURN HERE IF ERROR
017414  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
017416  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
017420  000137  017466                  JMP     25$             ;GO TO 25$ IF ERROR
1114 017424                     23$:
1115 017424  004737  041760            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
017430  000405                          BR      24$             ;GO TO 24$ IF NO ERROR
017432  000240                          NOP                     ;RETURN HERE IF ERROR
017434  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
017436  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
017440  000137  017466                  JMP     25$             ;GO TO 25$ IF ERROR
1116 017444                     24$:
1117 017444  032737  001000  001444    BIT     #SSEI,RMOFO     ;TEST 16 BIT MODE W/ SSEI SET ?
1118 017452  001005                    BNE     25$             ;YES !!
1119 017454  052737  001000  001444    BIS     #SSEI,RMOFO     ;SET SSEI IN OFFSET AND
1120 017462  000137  016474            JMP     2$              ;TEST AGAIN
1121 017466                     25$:
1122
1123                            ;;*****************************************************************
                                ;*TEST 11        WRITE, READ W/ IMPLIED SEEK
                                ;;*****************************************************************
017466                          TST11:
017466  000004                          SCOPE                   ;SCOPE CALL
017470  000240                          NOP                     ;START OF TEST
017472  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
017476  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
017502  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
017506  012737  000011  001226          MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1124
1125                            ;;*****************************************************************
                                ;LOOP #1        FORMAT,WRITE,READ
1126
1127
1128 017514                     1$:
1129
1130                            ;PREPARE THE DEVICE FOR FORMAT OPERATION
1131 017514  004737  034032            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
017520  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
017522  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
017524  000240                          NOP                     ;RETURN HERE IF ERROR
017526  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUPROUTINE
017530  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1132
1133                            ;LOAD PARAMETERS AND GENERATE DATA BUFFER
```

```
1134 017534                                    2$:
1135 017534  012737  001057  001446                MOV     #559.,RMDCO             ;CYLINDER = 559.
1136 017542  013737  001446  020710                MOV     RMDCO,32$               ;SAVE DESIRED CYLINDER
1137 017550  012737  001000  001420                MOV     #TA2,RMDAO              ;TRACK = 2, SECTOR = 0
1138 017556  012737  104254  001416                MOV     #BUFONE,RMBAO           ;BUS ADDRESS
1139 017564  012737  177376  001414                MOV     #-258.,RMWCO            ;2 * 256. WORDS (2'S COMP)
1140 017572  012737  010000  001444                MOV     #FMT16,RMOFO            ;16 BIT FORMAT
1141 017600  012737  000063  001412                MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
1142
1143                                           ;VERIFY THAT SECTOR IS NOT BAD
     017606  004737  034762                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     017612  000405                                BR      3$              ;GO TO 3$ IF NO ERROR
     017614  104401  066006                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     017620  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     017622  000137  020706                        JMP     31$             ;GO TO 31$ IF ERROR
1144 017626                                    3$:
1145 017626  012737  067414  001174                MOV     #ONES,$TMPO             ;STARTING ADDRESS OF PATTERN
1146 017634  012737  000001  001176                MOV     #1,$TMP1                ;RANGE OF PATTERN
1147 017642  004737  037214                        JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
1148
1149                                           ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
1150 017646  012702  001555                        MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
1151 017652  112722  000034                        MOVB    #RMDC,(R2)+
1152 017656  112722  000006                        MOVB    #RMDA,(R2)+
1153 017662  112722  000004                        MOVB    #RMBA,(R2)+
1154 017666  112722  000002                        MOVB    #RMWC,(R2)+
1155 017672  112722  000032                        MOVB    #RMOF,(R2)+
1156 017676  112722  000000                        MOVB    #RMCS1,(R2)+
1157 017702  112712  000200                        MOVB    #200,(R2)               ;TERMINATE TABLE
1158 017706                                    4$:
1159
1160                                           ;FORMAT THE DRIVE
1161 017706  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     017712  000404                                BR      5$              ;GO TO 5$ IF NO ERROR
     017714  000240                                NOP                     ;RETURN HERE IF ERROR
     017716  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     017720  000137  020706                        JMP     31$             ;GO TO 31$ IF ERROR
1162 017724                                    5$:
1163
1164                                           ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     017724  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1165
1166                                           ;WAIT FOR COMMAND TO COMPLETE
     017730  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1167
1168                                           ;GO READ STATUS FOR FORMAT OPERATION
1169 017734  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     017740  000404                                BR      6$              ;GO TO 6$ IF NO ERROR
     017742  000240                                NOP                     ;RETURN HERE IF ERROR
     017744  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     017746  000137  020706                        JMP     31$             ;GO TO 31$ IF ERROR
1170 017752                                    6$:
1171
1172                                           ;VERIFY NO ERRORS DURING FORMAT
1173 017752  004737  054166                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     017756  000405                                BR      7$              ;GO TO 7$ IF NO ERROR
     017760  000240                                NOP                     ;RETURN HERE IF ERROR
```

T11 WRITE, READ W/ IMPLIED SEEK

```
        017762  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        017764  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017766  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1174    017772                          7$:
1175
1176                                    ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1177    017772  012737  020002  001124          MOV     #8$,$LPERR
1178    020000  000410                          BR      9$                      ;SKIP TO WRITE OPERATION
1179
1180                                    ;;*****************************************************************
1181                                    ;LOOP #2        WRITE,READ
1182
1183    020002                          8$:
1184
1185                                    ;PREPARE DEVICE FOR WRITE OPERATION
1186    020002  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        020006  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        020010  000404                          BR      9$              ;GO TO 9$ IF NO ERROR
        020012  000240                          NOP                     ;RETURN HERE IF ERROR
        020014  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        020016  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1187    020022                          9$:
1188    020022  012737  000000  001446          MOV     #0,RMDCO                ;SEEK TO FIRST CYLINDER
1189    020030  012737  000005  001412          MOV     #SEEK!GO,RMCS10         ;WRITE SEEK COMMAND
1190
1191    020036  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        020042  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
        020044  000240                          NOP                     ;RETURN HERE IF ERROR
        020046  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        020050  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1192    020054                          10$:
1193
1194                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        020054  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1195
1196                                    ;WAIT FOR COMMAND TO COMPLETE
        020060  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1197
1198                                    ;GO GET REGISTER STATUS
1199    020064  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        020070  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
        020072  000240                          NOP                     ;RETURN HERE IF ERROR
        020074  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        020076  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1200    020102                          11$:
1201
1202                                    ;VERIFY RESULTS OF SEEK COMMAND
1203    020102  004737  046550                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        020106  000405                          BR      12$             ;GO TO 12$ IF NO ERROR
        020110  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        020112  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        020114  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020116  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
  1204  020122                          12$:
  1205
  1206                                  ;SETUP PARAMETERS AND EXCUTE WRITE DATA COMMAND
  1207  020122                          13$:
  1208  020122  013737  020710  001446          MOV     32$,RMDC0               ;RESTORE CYLINDER
  1209  020130  012737  177400  001414          MOV     #-256.,RMWC0            ;CHANGE WORD COUNT
  1210  020136  012737  104260  001416          MOV     #BUF0NE+4,RMBA0         ;CHANGE BUS ADDRESS
  1211  020144  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
  1212  020152  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  1213  020156  112722  000006                  MOVB    #RMDA,(R2)+
  1214  020162  112722  000034                  MOVB    #RMDC,(R2)+
  1215  020166  112722  000032                  MOVB    #RMOF,(R2)+
  1216  020172  112722  000004                  MOVB    #RMBA,(R2)+
  1217  020176  112722  000002                  MOVB    #RMWC,(R2)+
  1218  020202  112722  000000                  MOVB    #RMCS1,(R2)+
  1219  020206  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
  1220
  1221  020212  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        020216  000404                          BR      14$             ;GO TO 14$ IF NO ERROR
        020220  000240                          NOP                     ;RETURN HERE IF ERROR
        020222  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        020224  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
  1222  020230                          14$:
  1223
  1224                                  ;WAIT FOR COMMAND TO COMPLETE
        020230  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  1225
  1226                                  ;GO READ STATUS FOR WRITE COMMAND
  1227  020234  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        020240  000404                          BR      15$             ;GO TO 15$ IF NO ERROR
        020242  000240                          NOP                     ;RETURN HERE IF ERROR
        020244  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        020246  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
  1228  020252                          15$:
  1229
  1230                                  ;CHECK FOR ERRORS DURING WRITE OPERATION
  1231  020252  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        020256  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        020260  000240                          NOP                     ;RETURN HERE IF ERROR
        020262  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        020264  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020266  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
  1232  020272                          16$:
  1233  020272  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        020276  000405                          BR      17$             ;GO TO 17$ IF NO ERROR
        020300  000240                          NOP                     ;RETURN HERE IF ERROR
        020302  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        020304  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020306  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
  1234  020312                          17$:
  1235  020312  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        020316  000405                          BR      18$             ;GO TO 18$ IF NO ERROR
        020320  000240                          NOP                     ;RETURN HERE IF ERROR
        020322  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
```

T11 WRITE, READ W/ IMPLIED SEEK

```
020324  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
020326  000137  020706                   JMP     31$             ;GO TO 31$ IF ERROR
1236 020332                     18$:
1237
1238                            ;CHANGE LOOP ADDRESSES
1239 020332  012737  020342  001124     MOV     #19$,$LPERR
1240 020340  000410                     BR      20$                     ;SKIP TO NEXT OPERATION
1241
1242                            ;;*****************************************************************
1243                            ;LOOP #3        READ
1244
1245 020342                     19$:
1246
1247                            ;PREPARE DEVICE FOR READ OPERATION
1248 020342  004737  034032             JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
020346  154130                         .WORD   15413C          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
020350  000404                         BR      20$             ;GO TO 20$ IF NO ERROR
020352  000240                         NOP                     ;RETURN HERE IF ERROR
020354  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
020356  000137  020706                 JMP     31$             ;GO TO 31$ IF ERROR
1249 020362                     20$:
1250 020362  012737  000000  001446     MOV     #0,RMDCO                ;SEEK TO FIRST CYLINDER
1251 020370  012737  000005  001412     MOV     #SEEK!GO,RMCS10         ;WRITE SEEK COMMAND
1252
1253 020376  004737  040360             JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
020402  000404                         BR      21$             ;GO TO 21$ IF NO ERROR
020404  000240                         NOP                     ;RETURN HERE IF ERROR
020406  104000                         EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
020410  000137  020706                 JMP     31$             ;GO TO 31$ IF ERROR
1254 020414                     21$:
1255
1256                            ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
020414  004737  040024             JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1257
1258                            ;WAIT FOR COMMAND TO COMPLETE
020420  004737  040732             JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1259
1260                            ;GO GET REGISTER STATUS
1261 020424  004737  040110             JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
020430  000404                         BR      22$             ;GO TO 22$ IF NO ERROR
020432  000240                         NOP                     ;RETURN HERE IF ERROR
020434  104000                         EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
020436  000137  020706                 JMP     31$             ;GO TO 31$ IF ERROR
1262 020442                     22$:
1263
1264                            ;VERIFY RESUTS OF SEEK COMMAND
1265 020442  004737  046550             JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
020446  000405                         BR      23$             ;GO TO 23$ IF NO ERROR
020450  000240                         NOP                     ;RETURN HERE IF ERROR
020452  104000                         EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
```

```
020454  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
020456  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1266
1267 020462                     23$:
1268
1269                            ;SETUP PARAMETERS AND EXCUTE READ DATA COMMAND
1270 020462                     24$:
1271 020462  013737  020710  001446      MOV     32$,RMDC0               ;RESTORE CYLINDER
1272 020470  012737  000071  001412      MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
1273 020476  012737  105264  001416      MOV     #BUFTWO+4,RMBA0         ;LOAD STARTING BUFFER ADDRESS
1274 020504  012702  001555              MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1275 020510  112722  000006              MOVB    #RMDA,(R2)+
1276 020514  112722  000032              MOVB    #RMOF,(R2)+
1277 020520  112722  000034              MOVB    #RMDC,(R2)+
1278 020524  112722  000004              MOVB    #RMBA,(R2)+
1279 020530  112722  000002              MOVB    #RMWC,(R2)+
1280 020534  112722  000000              MOVB    #RMCS1,(R2)+
1281 020540  112712  000200              MOVB    #200,(R2)
1282
1283 020544  004737  040360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
020550  000404                          BR      25$             ;GO TO 25$ IF NO ERROR
020552  000240                          NOP                     ;RETURN HERE IF ERROR
020554  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
020556  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1284 020562                     25$:
1285
1286                            ;WAIT FOR COMMAND TO COMPLETE
020562  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1287
1288                            ;GO READ STATUS FOR READ OPERATION
1289 020566  004737  040110              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
020572  000404                          BR      26$             ;GO TO 26$ IF NO ERROR
020574  000240                          NOP                     ;RETURN HERE IF ERROR
020576  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
020600  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1290 020604                     26$:
1291
1292                            ;CHECK FOR ERRORS DURING READ OPERATION
1293 020604  004737  041126              JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
020610  000405                          BR      27$             ;GO TO 27$ IF NO ERROR
020612  000240                          NOP                     ;RETURN HERE IF ERROR
020614  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
020616  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
020620  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1294 020624                     27$:
1295 020624  004737  054166              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
020630  000405                          BR      28$             ;GO TO 28$ IF NO ERROR
020632  000240                          NOP                     ;RETURN HERE IF ERROR
020634  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
020636  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
020640  000137  020706                  JMP     31$             ;GO TO 31$ IF ERROR
1296 020644                     28$:
1297 020644  004737  041760              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
020650  000405                          BR      29$             ;GO TO 29$ IF NO ERROR
020652  000240                          NOP                     ;RETURN HERE IF ERROR
020654  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
020656  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
T11     WRITE, READ W/ IMPLIED SEEK

        020660  000137  020706                 JMP     31$             ;GO TO 31$ IF ERROR
  1298  020664                          29$:
  1299  020664  004737  037462                 JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
        020670  104260                         .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
        020672  105264                         .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
        020674  000404                         BR      30$             ;GO TO 30$ IF NO ERROR
        020676  000240                         NOP                     ;RETURN HERE IF ERROR
        020700  104000                         EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        020702  000137  020706                 JMP     31$             ;GO TO 31$ IF ERROR
  1300  020706                          30$:
  1301
  1302  020706                          31$:
  1303  020706  000401                         BR      33$
  1304
  1305  020710  000000                  32$:   .WORD   0               ;STORAGE FOR DATA CYLINDER
  1306
  1307  020712                          33$:
  1308
  1309                                  ;;*******************************************************************
                                        ;*TEST 12        READ, WRITE CHECK W/ HEAD SWITCHING
                                        ;;*******************************************************************
        020712                          TST12:
        020712  000004                         SCOPE                   ;SCOPE CALL
        020714  000240                         NOP                     ;START OF TEST
        020716  012706  001100                 MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        020722  013700  001276                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        020726  013701  001466                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        020732  012737  000012  001226         MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  1310
  1311                                  ;;*******************************************************************
  1312                                  ;LOOP #1        READ,WRITE CHECK
  1313
  1314                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  1315  020740                          1$:
  1316  020740  012737  001036  001420         MOV     #TA2!30.,RMDA0          ;TRACK = 2, SECTOR = 30.
  1317  020746  012737  010000  001444         MOV     #FMT16,RMOF0            ;16 BIT FORMAT
  1318  020754                          2$:
  1319  020754  012737  001057  001446         MOV     #559.,RMDC0             ;CYLINDER = 559.
  1320  020762  012737  104254  001416         MOV     #BUFONE,RMBA0           ;BUS ADDRESS
  1321  020770  012737  177000  001414         MOV     #-256.*2,RMWC0          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
  1322  020776  012737  000070  001412         MOV     #RD,RMCS10              ;READ DATA COMMAND
  1330  021004                          3$:
  1331
  1332                                  ;PREPARE THE DEVICE FOR FORMAT OPERATION
  1333  021004  004737  034032                 JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        021010  154130                         .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        021012  000404                         BR      4$              ;GO TO 4$ IF NO ERROR
        021014  000240                         NOP                     ;RETURN HERE IF ERROR
        021016  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        021020  000137  021734                 JMP     26$             ;GO TO 26$ IF ERROR
```

```
1334 021024                                4$:
1335
1336                                       ;SETUP PARAMETERS AND SEEK TO GET DRIVE ON CYLINDER
1337 021024  012737  000005  001412              MOV     #SEEK!GO,RMCS1O          ;LOAD SEEK COMMAND
1338 021032  012702  001555                      MOV     #PUTINX,R2               ;R2 = BYTE ENTRY POSITION
1339 021036  112722  000034                      MOVB    #RMDC,(R2)+
1340 021042  112722  000006                      MOVB    #RMDA,(R2)+
1341 021046  112722  000004                      MOVB    #RMBA,(R2)+
1342 021052  112722  000002                      MOVB    #RMWC,(R2)+
1343 021056  112722  000032                      MOVB    #RMOF,(R2)+
1344 021062  112722  000000                      MOVB    #RMCS1,(R2)+
1345 021066  112712  000200                      MOVB    #200,(R2)                ;TERMINATE TABLE
1346 021072                                5$:
1347
1348 021072  004737  040360                      JSR     PC,PUT           ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021076  000404                              BR      6$               ;GO TO 6$ IF NO ERROR
     021100  000240                              NOP                      ;RETURN HERE IF ERROR
     021102  104000                              EMT                      ;ERROR # DEFINED BY PUT SUBROUTINE
     021104  000137  021734                      JMP     26$              ;GO TO 26$ IF ERROR
1349 021110                                6$:
1350
1351                                       ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     021110  004737  040024                      JSR     PC,GETSTS        ;GO TO GETSTS SUBROUTINE
1352
1353                                       ;WAIT FOR COMMAND TO COMPLETE
     021114  004737  040732                      JSR     PC,TIMOUT        ;GO TO TIMOUT SUBROUTINE
1354
1355                                       ;GO READ REGISTER STATUS
1356 021120  004737  040110                      JSR     PC,GET           ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021124  000404                              BR      7$               ;GO TO 7$ IF NO ERROR
     021126  000240                              NOP                      ;RETURN HERE IF ERROR
     021130  104000                              EMT                      ;ERROR # DEFINED BY GET SUBROUTINE
     021132  000137  021734                      JMP     26$              ;GO TO 26$ IF ERROR
1357 021136                                7$:
1358
1359                                       ;GO VERIFY RESULTS OF SEEK
1360 021136  004737  046550                      JSR     PC,SEKSTS        ;GO VERIFY RESULTS OF SEEK OPERATION
     021142  000405                              BR      8$               ;GO TO 8$ IF NO ERROR
     021144  000240                              NOP                      ;RETURN HERE IF ERROR
     021146  104000                              EMT                      ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     021150  004736                              JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
     021152  000137  021734                      JMP     26$              ;GO TO 26$ IF ERROR
1361 021156                                8$:
1362
1363                                       ;READ DATA FROM THE DRIVE
1364 021156                                9$:
1365 021156  012737  000071  001412              MOV     #RD!GO,RMCS1O            ;READ DATA COMMAND
1369 021164  012702  001555                      MOV     #PUTINX,R2               ;LOAD PUT REGISTER INDEX TABLE
1370 021170  112722  000006                      MOVB    #RMDA,(R2)+
1371 021174  112722  000034                      MOVB    #RMDC,(R2)+
1372 021200  112722  000032                      MOVB    #RMOF,(R2)+
1373 021204  112722  000004                      MOVB    #RMBA,(R2)+
1374 021210  112722  000002                      MOVB    #RMWC,(R2)+
1375 021214  112722  000000                      MOVB    #RMCS1,(R2)+
1376 021220  112722  000200                      MOVB    #200,(R2)+               ;TERMINATE TABLE
1377
1378 021224  004737  040360                      JSR     PC,PUT           ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
```

```
021230  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
021232  000240                          NOP                     ;RETURN HERE IF ERROR
021234  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
021236  000137  021734                  JMP     26$             ;GO TO 26$ IF ERROR
1379 021242                     10$:
1380
1381                            ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
021242  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1382
1383                            ;WAIT FOR COMMAND TO COMPLETE
021246  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1384
1385                            ;GO READ STATUS FOR READ COMMAND
1386 021252  004737  040110              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
021256  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
021260  000240                          NOP                     ;RETURN HERE IF ERROR
021262  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
021264  000137  021734                  JMP     26$             ;GO TO 26$ IF ERROR
1387
1388                            ;CHECK TO SEE IF THIS SECTOR WAS SKIP SECTORED FIRST, IF SO
1389                            ;DO NOT EXECUTE THIS TEST IN 16 BIT FORMAT MODE
1390 021270                     11$:
1391 021270  032737  000040  001400      BIT     #SSE,RMER2I             ;IS 'SSE' BIT SET ?
1392 021276  001030                      BNE     14$                     ;BR IF YES
1393
1394                            ;CHECK FOR ERRORS DURING READ OPERATION
1395 021300  004737  041126              JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
021304  000405                          BR      12$             ;GO TO 12$ IF NO ERROR
021306  000240                          NOP                     ;RETURN HERE IF ERROR
021310  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
021312  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021314  000137  021734                  JMP     26$             ;GO TO 26$ IF ERROR
1396 021320                     12$:
1397 021320  004737  054166              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
021324  000405                          BR      13$             ;GO TO 13$ IF NO ERROR
021326  000240                          NOP                     ;RETURN HERE IF ERROR
021330  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
021332  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021334  000137  021734                  JMP     26$             ;GO TO 26$ IF ERROR
1398 021340                     13$:
1399 021340  004737  041760              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
021344  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
021346  000240                          NOP                     ;RETURN HERE IF ERROR
021350  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
021352  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021354  000137  021734                  JMP     26$             ;GO TO 26$ IF ERROR
1400 021360                     14$:
:401
1402                            ;CHANGE LOOP ADDRESSES
1403 021360  012737  021370  001124      MOV     #15$,$LPERR
1404 021366  000410                      BR      16$                     ;SKIP TO NEXT OPERATION
1405
1406                            ;;*****************************************************************
1407                            ;LOOP #2        WRITE CHECK DATA
1408
1409 021370                     15$:
1410
```

```
1411                                    ;PREPARE DEVICE FOR READ OPERATION
1412 021370  004737  034032                     JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     021374  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
     021376  000404                             BR      16$             ;GO TO 16$ IF NO ERROR
     021400  000240                             NOP                     ;RETURN HERE IF ERROR
     021402  104000                             EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     021404  000137  021734                     JMP     26$             ;GO TO 26$ IF ERROR
:413 021410                             16$:
1414 021410  012737  000005  001412            MOV     #SEEK!GO,RMCS10         ;LOAD SEEK COMMAND
1415
1416 021416  004737  040360                     JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021422  000404                             BR      17$             ;GO TO 17$ IF NO ERROR
     021424  000240                             NOP                     ;RETURN HERE IF ERROR
     021426  104000                             EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021430  000137  021734                     JMP     26$             ;GO TO 26$ IF ERROR
1417 021434                             17$:
1418
1419                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     021434  004737  040024                     JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1420
1421                                    ;WAIT FOR COMMAND TO COMPLETE
     021440  004737  040732                     JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1422
1423                                    ;GO READ REGISTER STATUS
1424 021444  004737  040110                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021450  000404                             BR      18$             ;GO TO 18$ IF NO ERROR
     021452  000240                             NOP                     ;RETURN HERE IF ERROR
     021454  104000                             EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021456  000137  021734                     JMP     26$             ;GO TO 26$ IF ERROR
1425 021462                             18$:
1426
1427                                    ;GO VERIFY RESLUTS OF SEEK
1428 021462  004737  046550                     JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     021466  000405                             BR      19$             ;GO TO 19$ IF NO ERROR
     021470  000240                             NOP                     ;RETURN HERE IF ERROR
     021472  104000                             EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     021474  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021476  000137  021734                     JMP     26$             ;GO TO 26$ IF ERROR
1429 021502                             19$:
1430
1431                                    ;READ DATA FROM DEVICE
1432 021502                             20$:
1433 021502  012737  000051  001412            MOV     #WCD!GO,RMCS10          ;READ DATA COMMAND
1434 021510  012702  001555                     MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1435 021514  112722  000006                     MOVB    #RMDA,(R2)+
1436 021520  112722  000032                     MOVB    #RMOF,(R2)+
1437 021524  112722  000034                     MOVB    #RMDC,(R2)+
1438 021530  112722  000004                     MOVB    #RMBA,(R2)+
1439 021534  112722  000002                     MOVB    #RMWC,(R2)+
1440 021540  112722  000000                     MOVB    #RMCS1,(R2)+
```

```
1441 021544  112712  000200                    MOVB    #200,(R2)
1442
1443 021550  004737  040360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     021554  000404                            BR      21$             ;GO TO 21$ IF NO ERROR
     021556  000240                            NOP                     ;RETURN HERE IF ERROR
     021560  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     021562  000137  021734                    JMP     26$             ;GO TO 26$ IF ERROR
1444 021566                                21$:
1445
1446                                       ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     021566  004737  040024                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1447
1448                                       ;WAIT FOR COMMAND TO COMPLETE
     021572  004737  040732                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1449
1450                                       ;GO READ STATUS FOR WRITE CHECK OPERATION
1451 021576  004737  040110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     021602  000404                            BR      22$             ;GO TO 22$ IF NO ERROR
     021604  000240                            NOP                     ;RETURN HERE IF ERROR
     021606  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     021610  000137  021734                    JMP     26$             ;GO TO 26$ IF ERROR
1452
1453                                       ;CHECK TO SEE IF THIS SECTOR WAS SKIP SECTORED FIRST, IF SO
1454                                       ;DO NOT EXECUTE THIS TEST IN 16 BIT FORMAT MODE
1455 021614                                22$:
1456 021614  032737  000040  001400            BIT     #SSE,RMER2I             ;IS 'SSE' BIT SET ?
1457 021622  001030                            BNE     25$                     ;BR IF YES
1458
1459                                       ;CHECK FOR ERRORS DURING WRITE CHECK OPERATION
1460 021624  004737  041126                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     021630  000405                            BR      23$             ;GO TO 23$ IF NO ERROR
     021632  000240                            NOP                     ;RETURN HERE IF ERROR
     021634  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     021636  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021640  000137  021734                    JMP     26$             ;GO TO 26$ IF ERROR
1461 021644                                23$:
1462 021644  004737  054166                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     021650  000405                            BR      24$             ;GO TO 24$ IF NO ERROR
     021652  000240                            NOP                     ;RETURN HERE IF ERROR
     021654  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     021656  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021660  000137  021734                    JMP     26$             ;GO TO 26$ IF ERROR
1463 021664                                24$:
1464 021664  004737  041760                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     021670  000405                            BR      25$             ;GO TO 25$ IF NO ERROR
     021672  000240                            NOP                     ;RETURN HERE IF ERROR
     021674  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     021676  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     021700  000137  021734                    JMP     26$             ;GO TO 26$ IF ERROR
1465 021704                                25$:
1466 021704  032737  001000  001444            BIT     #SSEI,RMOFO     ;TEST 16 BIT MODE W/ SSEI SET ?
1467 021712  001010                            BNE     26$             ;YES !!
1468 021714  012737  001037  001420            MOV     #TA2!31.,RMDAO  ;TRACK = 1, SECTOR = 31.
1469 021722  052737  001000  001444            BIS     #SSEI,RMOFO     ;SET SSEI BIT IN OFFSET AND
1470 021730  000137  020754                    JMP     2$              ;TEST AGAIN.
1471 021734                                26$:
1472
```

```
1473                                    ;;*****************************************************************
                                        ;*TEST 13        READ, WRITE CHECK W/ MID-TRANSFER SEEK
                                        ;;*****************************************************************
        021734                          TST13:
        021734  000004                          SCOPE                   ;SCOPE CALL
        021736  000240                          NOP                     ;START OF TEST
        021740  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        021744  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        021750  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        021754  012737  000013  001226          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1474
1475                                    ;;*****************************************************************
1476                                    ;LOOP #1        READ,WRITE CHECK
1477
1478                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1479 021762                            1$:
1480 021762  013737  001334  001420          MOV     LSTRK,RMDAO             ;SET LAST TRACK AND
1481 021770  112737  000036  001420          MOVB    #30.,RMDAO              ;LAST SECTOR
1482 021776  012737  010000  001444          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
1483 022004                            2$:
1484 022004  012737  001057  001446          MOV     #559.,RMDCO             ;CYLINDER = 559.
1485 022012  012737  177000  001414          MOV     #-256.*2,RMWCO          ;CHANGE WORD COUNT
1486 022020  012737  104254  001416          MOV     #BUFONE,RMBAO           ;CHANGE BUS ADDRSS
1487 022026  012737  000070  001412          MOV     #RD,RMCS1O              ;READ DATA COMMAND
1495 022034                            3$:
1496
1497                                    ;PREPARE DEVICE FOR READ OPERATION
1498 022034  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        022040  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        022042  000404                          BR      4$              ;GO TO 4$ IF NO ERROR
        022044  000240                          NOP                     ;RETURN HERE IF ERROR
        022046  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        022050  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1499 022054                            4$:
1500
1501                                    ;READ DATA FROM THE DRIVE
1502 022054                            5$:
1503 022054  012737  000071  001412          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
1507 022062  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
1508 022066  112722  000006                  MOVB    #RMDA,(R2)+
1509 022072  112722  000034                  MOVB    #RMDC,(R2)+
1510 022076  112722  000032                  MOVB    #RMOF,(R2)+
1511 022102  112722  000004                  MOVB    #RMBA,(R2)+
1512 022106  112722  000002                  MOVB    #RMWC,(R2)+
1513 022112  112722  000000                  MOVB    #RMCS1,(R2)+
1514 022116  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
1515
1516 022122  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        022126  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        022130  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        022132  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        022134  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1517    022140                          6$:
1518
1519                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        022140  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1520
1521                                    ;WAIT FOR COMMAND TO COMPLETE
        022144  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1522
1523                                    ;GO READ STATUS FOR READ COMMAND
1524    022150  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        022154  000404                          BR      7$              ;GO TO 7$ IF NO ERROR
        022156  000240                          NOP                     ;RETURN HERE IF ERROR
        022160  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        022162  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1525
1526                                    ;CHECK TO SEE IF THIS SECTOR WAS SKIP SECTORED FIRST, IF SO
1527                                    ;DO NOT EXECUTE THIS TEST IN 16 BIT FORMAT MODE
1528    022166                          7$:
1529    022166  032737  000040  001400          BIT     #SSE,RMER2I             ;IS 'SSE' BIT SET ?
1530    022174  001030                          BNE     10$                     ;BP IF YES
1531
1532                                    ;CHECK FOR ERRORS DURING READ OPERATION
1533    022176  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        022202  000405                          BR      8$              ;GO TO 8$ IF NO ERROR
        022204  000240                          NOP                     ;RETURN HERE IF ERROR
        022206  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        022210  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022212  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1534    022216                          8$:
1535    022216  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        022222  000405                          BR      9$              ;GO TO 9$ IF NO ERROR
        022224  000240                          NOP                     ;RETURN HERE IF ERROR
        022226  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        022230  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022232  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1536    022236                          9$:
1537    022236  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        022242  000405                          BR      10$             ;GO TO 10$ IF NO ERROR
        022244  000240                          NOP                     ;RETURN HERE IF ERROR
        022246  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        022250  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022252  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1538    022256                          10$:
1539
1540                                    ;CHANGE LOOP ADDRESSES
1541    022256  012737  022266  001124          MOV     #11$,$LPERR
1542    022264  000410                          BR      12$                     ;SKIP TO NEXT OPERATION
1543
1544                                    ;;*****************************************************************
1545                                    ;LOOP #2        WRITE CHECK DATA
1546
1547    022266                          11$:
1548
1549                                    ;PREPARE DEVICE FOR WRITE CHECK OPERATION
1550    022266  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
```

```
        022272  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
        022274  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
        022276  000240                          NOP                     ;RETURN HERE IF ERROR
        022300  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        022302  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
  1551  022306                          12$:
  1552
  1553                                  ;WRITE CHECK DATA FROM DEVICE
  1554  022306                          13$:
  1555  022306  012737  000051  001412          MOV     #WCD!GO,RMCS1O          ;READ DATA COMMAND
  1556  022314  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
  1557  022320  112722  000006                  MOVB    #RMDA,(R2)+
  1558  022324  112722  000032                  MOVB    #RMOF,(R2)+
  1559  022330  112722  000034                  MOVB    #RMDC,(R2)+
  1560  022334  112722  000004                  MOVB    #RMBA,(R2)+
  1561  022340  112722  000002                  MOVB    #RMWC,(R2)+
  1562  022344  112722  000000                  MOVB    #RMCS1,(R2)+
  1563  022350  112712  000200                  MOVB    #200,(R2)
  1564
  1565  022354  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        022360  000404                          BR      14$             ;GO TO 14$ IF NO ERROR
        022362  000240                          NOP                     ;RETURN HERE IF ERROR
        022364  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        022366  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
  1566  022372                          14$:
  1567
  1568                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
        022372  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
  1569
  1570                                  ;WAIT FOR COMMAND TO COMPLETE
        022376  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
  1571
  1572                                  ;GO READ STATUS FOR READ OPERATION
  1573  022402  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        022406  000404                          BR      15$             ;GO TO 15$ IF NO ERROR
        022410  000240                          NOP                     ;RETURN HERE IF ERROR
        022412  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        022414  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
  1574
  1575                                  ;CHECK TO SEE IF THIS SECTOR WAS SKIP SECTORED FIRST, IF SO
  1576                                  ;DO NOT EXECUTE THIS TEST IN 16 BIT FORMAT MODE
  1577  022420                          15$:
  1578  022420  032737  000040  001400          BIT     #SSE,RMER2I             ;IS 'SSE' BIT SET ?
  1579  022426  001030                          BNE     18$                     ;BR IF YES
  1580
  1581                                  ;CHECK FOR ERRORS DURING READ OPERATION
  1582  022430  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        022434  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        022436  000240                          NOP                     ;RETURN HERE IF ERROR
        022440  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
```

```
                022442  004736                          JSR     PC,@(SP)+       ;GO PACK FOR MORE ERROR CHECKS
                022444  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1583            022450                          16$:
1584            022450  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                022454  000405                          BR      17$             ;GO TO 17$ IF NO ERROR
                022456  000240                          NOP                     ;RETURN HERE IF ERROR
                022460  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                022462  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022464  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1585            022470                          17$:
1586            022470  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                022474  000405                          BR      18$             ;GO TO 18$ IF NO ERROR
                022476  000240                          NOP                     ;RETURN HERE IF ERROR
                022500  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                022502  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022504  000137  022546                  JMP     19$             ;GO TO 19$ IF ERROR
1587            022510                          18$:
1588            022510  032737  001000  001444          BIT     #SSEI,RMOFO     ;TEST 16 BIT MODE W/ SSEI SET ?
1589            022516  001013                          BNE     19$             ;YES !!
1590            022520  013737  001334  001420          MOV     LSTRK,RMDAO     ;SET LAST TRACK AND
1591            022526  112737  000037  001420          MOVB    #31.,RMDAO      ;LAST SECTOR
1592            022534  052737  001000  001444          BIS     #SSEI,RMOFO     ;SET SSEI BIT IN RMOF AND
1593            022542  000137  022004                  JMP     2$              ;TEST AGAIN.
1594            022546                          19$:
1595
1596                                            ;;*******************************************************************
                                                ;*TEST 14        WRITE, READ W/ HCE ERROR
                                                ;;*******************************************************************
                022546                          ST14:
                022546  000004                          SCOPE                   ;SCOPE CALL
                022550  000240                          NOP                     ;START OF TEST
                022552  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                022556  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                022562  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                022566  012737  000014  001226          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1597
1598                                            ;;*******************************************************************
1599                                            ;LOOP #1        FORMAT,WRITE,READ
1600
1601            022574  012704  000000                  MOV     #0,R4           ;R4 = 1ST HEADER WORD
1602            022600  012703  000001          1$:     MOV     #1,R3           ;R3 = HCE BIT
1603            022604                          2$:
1604
1605                                            ;PREPARE THE DEVICE FOR FORMAT OPERATION
1606            022604  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                022610  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
                022612  000404                          BR      3$              ;GO TO 3$ IF NO ERROR
                022614  000240                          NOP                     ;RETURN HERE IF ERROR
                022616  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                022620  000137  024416                  JMP     47$             ;GO TO 47$ IF ERROR
```

```
1607
1608                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1609 022624                                 3$:
1610 022624  012737  001057  001446                MOV     #559.,RMDCO     ;CYLINDER = 559.
1611 022632  012737  001000  001420                MOV     #TA2,RMDAO      ;TRACK = 2, SECTOR = 0
1612 022640  012737  104254  001416                MOV     #BUFONE,RMBAO   ;BUS ADDRESS
1613 022646  012737  177376  001414                MOV     #-258.,RMWCO    ;2 + 256. WORDS (2'S COMP)
1614 022654  012737  010000  001444                MOV     #FMT16,RMOFO    ;16 BIT FORMAT
1615 022662  012737  000063  001412                MOV     #WH!GO,RMCS1O   ;WRITE HEADER AND DATA COMMAND
1616
1617                                        ;VERIFY THAT SECTOR IS NOT BAD
     022670  004737  034762                        JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     022674  000405                                BR      4$              ;GO TO 4$ IF NO ERROR
     022676  104401  066006                        TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     022702  104000                                EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     022704  000137  024416                        JMP     46$             ;GO TO 46$ IF ERROR
1618 022710                                 4$:
1619 022710  012737  067456  001174                MOV     #ZEROS,$TMPO    ;STARTING ADDRESS OF PATTERN
1620 022716  012737  000001  001176                MOV     #1,$TMP1        ;RANGE OF PATTERN
1621 022724  004737  037214                        JSR     PC,GENBUF       ;GO GENERATE BUFFER FOR FORMAT
1622
1623                                        ;CHANGE HEADER WORD TO FORCE HCE DURING WRITE & READ
1624 022730  030364  104254                        BIT     R3,BUFONE(R4)   ;SET OR RESET FOR HCE??
1625 022734  001403                                BEQ     5$
1626 022736  040364  104254                        BIC     R3,BUFONE(R4)   ;RESET BIT FOR HCE
1627 022742  000402                                BR      6$
1628 022744  050364  104254                 5$:    BIS     R3,BUFONE(R4)   ;SET FOR HCE
1629
1630                                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
1631 022750                                 6$:
1632 022750  012702  001555                        MOV     #PUTINX,R2      ;R2 = BYTE ENTRY POSITION
1633 022754  112722  000034                        MOVB    #RMDC,(R2)+
1634 022760  112722  000006                        MOVB    #RMDA,(R2)+
1635 022764  112722  000004                        MOVB    #RMBA,(R2)+
1636 022770  112722  000002                        MOVB    #RMWC,(R2)+
1637 022774  112722  000032                        MOVB    #RMOF,(R2)+
1638 023000  112722  000000                        MOVB    #RMCS1,(R2)+
1639 023004  112712  000200                        MOVB    #200,(R2)       ;TERMINATE TABLE
1640 023010                                 7$:
1641 023010  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     023014  000404                                BR      8$              ;GO TO 8$ IF NO ERROR
     023016  000240                                NOP                     ;RETURN HERE IF ERROR
     023020  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     023022  000137  024416                        JMP     46$             ;GO TO 46$ IF ERROR
1642 023026                                 8$:
1643
1644                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     023026  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1645
1646                                        ;WAIT FOR COMMAND TO COMPLETE
     023032  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1647
1648                                        ;GO READ STATUS FOR FORMAT OPERATION
1649 023036  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     023042  000404                                BR      9$              ;GO TO 9$ IF NO ERROR
     023044  000240                                NOP                     ;RETURN HERE IF ERROR
     023046  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
```

```
023050  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1650 023054                     9$:
1651
1652                            ;VERIFY NO ERRORS DURING FORMAT
1653 023054  004737  054166              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
023060  000405                  BR      10$             ;GO TO 10$ IF NO ERROR
023062  000240                  NOP                     ;RETURN HERE IF ERROR
023064  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
023066  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
023070  000137  024416          JMP     46$             ;GO TO 46$ IF ERROR
1654 023074                     10$:
1655
1656                            ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1657 023074  012737  023104  001124      MOV     #11$,$LPERR
1658 023102  000410              BR      12$             ;SKIP TO WRITE OPERATION
1659
1660                            ;;*****************************************************************
1661                            ;LOOP #2        WRITE,READ
1662
1663 023104                     11$:
1664
1665                            ;PREPARE DEVICE FOR WRITE OPERATION
1666 023104  004737  034032              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
023110  154130                  .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                        ;VERIFY PACK ACKNOWLEDGE
                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                        ;VERIFY RECALIBRATION
023112  000404                  BR      12$             ;GO TO 12$ IF NO ERROR
023114  000240                  NOP                     ;RETURN HERE IF ERROR
023116  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
023120  000137  024416          JMP     46$             ;GO TO 46$ IF ERROR
1667 023124                     12$:
1668
1669                            ;WRITE DATA TO THE DRIVE
1670 023124                     13$:
1671 023124  012737  104260  001416      MOV     #BUFONE+4,RMBAO ;CHANGE BUS ADDRESS
1672 023132  012737  177400  001414      MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
1673 023140  012737  000061  001412      MOV     #WD!GO,RMCS10   ;WRITE DATA COMMAND
1674 023146  012702  001555              MOV     #PUTINX,R2      ;LOAD PUT REGISTER INDEX TABLE
1675 023152  112722  000006              MOVB    #RMDA,(R2)+
1676 023156  112722  000034              MOVB    #RMDC,(R2)+
1677 023162  112722  000032              MOVB    #RMOF,(R2)+
1678 023166  112722  000004              MOVB    #RMBA,(R2)+
1679 023172  112722  000002              MOVB    #RMWC,(R2)+
1680 023176  112722  000000              MOVB    #RMCS1,(R2)+
1681 023202  112722  000200              MOVB    #200,(R2)+      ;TERMINATE TABLE
1682
1683 023206  004737  040360              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
023212  000404                  BR      14$             ;GO TO 14$ IF NO ERROR
023214  000240                  NOP                     ;RETURN HERE IF ERROR
023216  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
023220  000137  024416          JMP     46$             ;GO TO 46$ IF ERROR
1684 023224                     14$:
```

```
1685
1686                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
023224  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1687
1688                                    ;WAIT FOR COMMAND TO COMPLETE
023230  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1689
1690                                    ;GO READ STATUS FOR WRITE COMMAND
1691 023234  004737  040110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     023240  000404                            BR      15$             ;GO TO 15$ IF NO ERROR
     023242  000240                            NOP                     ;RETURN HERE IF ERROR
     023244  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     023246  000137  024416                    JMP     46$             ;GO TO 46$ IF ERROR
1692 023252                             15$:
1693
1694                                    ;CHECK FOR ERRORS DURING WRITE OPERATION
1695 023252  004737  041126                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     023256  000405                            BR      16$             ;GO TO 16$ IF NO ERROR
     023260  000240                            NOP                     ;RETURN HERE IF ERROR
     023262  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     023264  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023266  000137  024416                    JMP     46$             ;GO TO 46$ IF ERROR
1696
1697                                    ;DECODE THE TYPE OF ERROR PRODUCED BY THE BIT POSITION OF R3
1698 023272                             16$:
1699 023272  005704                            TST     R4              ;IS THIS THE FIRST HEADER WORD ?
1700 023274  001011                            BNE     17$             ;NO !!
1701
1702 023276  032703  020000                    BIT     #SSF,R3         ;SHOULD SSE BE SET ?
1703 023302  001042                            BNE     20$             ;YES !!
1704 023304  032703  010000                    BIT     #FMT16,R3       ;SHOULD FER BE SET ?
1705 023310  001067                            BNE     22$             ;YES !!
1706 023312  032703  140000                    BIT     #MSE!USE,R3     ;SHOULD BSE BE SET ?
1707 023316  001111                            BNE     24$             ;YES !!
1708
1709                                    ;VERIFY THAT A HEADER COMPARE ERROR WAS DETECTED
1710 023320                             17$:
1711 023320  032737  000200  001352              BIT     #HCE,RMER1I     ;WAS HCE DETECTED ??
1712 023326  001132                            BNE     26$             ;YES !!
1713
1714 023330  004737  054166                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     023334  000405                            BR      18$             ;GO TO 18$ IF NO ERROR
     023336  000240                            NOP                     ;RETURN HERE IF ERROR
     023340  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     023342  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023344  000137  024416                    JMP     46$             ;GO TO 46$ IF ERROR
1715 023350                             18$:
1716 023350  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1717 023356  012737  000200  001140            MOV     #HCE,$GDDAT     ;EXPECTED STATUS
1718 023364  010437  001174                    MOV     R4,$TMP0        ;R4 = HEADER WORD NUMBER AND
1719 023370  001002                            BNE     19$             ;ADJUST NUMBER.
1720 023372  005237  001174                    INC     $TMP0
1721 023376  010337  001176            19$:    MOV     R3,$TMP1        ;R3 = BIT POSIITON
1722 023402  104344                            EMT     344
1723 023404  000137  024416                    JMP     46$
1724
1725                                    ;VERIFY THAT SSE IS SET
```

```
1726 023410                                    20$:
1727 023410  032737  000040  001400                   BIT     #SSE,RMER2I     ;IS SSE SET ?
1728 023416  001076                                   BNE     26$             ;YES !!
1729
1730 023420  004737  054166                           JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     023424  000405                                   BR      21$             ;GO TO 21$ IF NO ERROR
     023426  000240                                   NOP                     ;RETURN HERE IF ERROR
     023430  104000                                   EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     023432  004736                                   JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023434  000137  024416                           JMP     46$             ;GO TO 46$ IF ERROR
1731 023440                                    21$:
1732 023440  013737  001400  001142                   MOV     RMER2I,$BDDAT   ;GET RECIEVED STATUS
1733 023446  013737  001400  001140                   MOV     RMER2I,$GDDAT   ;GET EXPECTED STATUS AND
1734 023454  052737  000040  001140                   BIS     #SSE,$GDDAT     ;SET SSE.
1735 023462  104224                                   EMT     224
1736 023464  000137  024416                           JMP     46$
1737
1738                                           ;VERIFY THAT A FORMAT ERROR WAS DETECTED
1739 023470                                    22$:
1740 023470  032737  000020  001352                   BIT     #FER,RMER1I     ;WAS FER DETECTED ??
1741 023476  001046                                   BNE     26$             ;YES !!
1742
1743 023500  004737  054166                           JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     023504  000405                                   BR      23$             ;GO TO 23$ IF NO ERROR
     023506  000240                                   NOP                     ;RETURN HERE IF ERROR
     023510  104000                                   EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     023512  004736                                   JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023514  000137  024416                           JMP     46$             ;GO TO 46$ IF ERROR
1744 023520                                    23$:
1745 023520  012737  000020  001140                   MOV     #FER,$GDDAT     ;EXPECTED STATUS
1746 023526  013737  001352  001142                   MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1747 023534  104343                                   EMT     343
1748 023536  000137  024416                           JMP     46$
1749
1750                                           ;VERIFY THAT A BAD SECTOR ERROR WAS DETECTED
1751 023542                                    24$:
1752 023542  032737  100000  001400                   BIT     #BSE,RMER2I     ;WAS BSE DETECTED ??
1753 023550  001021                                   BNE     26$             ;YES !!
1754
1755 023552  004737  054166                           JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     023556  000405                                   BR      25$             ;GO TO 25$ IF NO ERROR
     023560  000240                                   NOP                     ;RETURN HERE IF ERROR
     023562  104000                                   EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     023564  004736                                   JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     023566  000137  024416                           JMP     46$             ;GO TO 46$ IF ERROR
1756 023572                                    25$:
1757 023572  013737  001400  001142                   MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
1758 023600  012737  100000  001140                   MOV     #BSE,$GDDAT     ;EXPECTED STAIUS
1759 023606  104345                                   EMT     345
1760 023610  000137  024416                           JMP     46$
1761
1762                                           ;CHECK FOR OTHER ERRORS
1763 023614                                    26$:
1764 023614  004737  041760                           JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     023620  000405                                   BR      27$             ;GO TO 27$ IF NO ERROR
     023622  000240                                   NOP                     ;RETURN HERE IF ERROR
     023624  104000                                   EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
```

```
023626  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
023630  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1765 023634                     27$:
1766
1767                            ;CHANGE LOOP ADDRESSES
1768 023634  012737  023642  001124     MOV     #28$,$LPERR
1769
1770                            ;;*****************************************************************
1771                            ;LOOP #3        READ
1772
1773 023642                     28$:
1774
1775                            ;PREPARE DEVICE FOR READ OPERATION
1776 023642  004737  034032             JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
023646  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 SE(
                                                                ;VERIFY RECALIBRATION
023650  000404                          BR      29$             ;GO TO 29$ IF NO ERROR
023652  000240                          NOP                     ;RETURN HERE IF ERROR
023654  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
023656  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1777 023662                     29$:
1778
1779                            ;READ DATA FROM DEVICE
1780 023662                     30$:
1781 023662  012737  105264  001416     MOV     #BUFTWO+4,RMBAO ;CHANGE BUS ADDRESS
1782 023670  012737  177400  001414     MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
1783 023676  012737  000071  001412     MOV     #RD!GO,RMCS10   ;READ DATA COMMAND
1784 023704  012702  001555             MOV     #PUTINX,R2      ;LOAD PUT REGISTER INDEX TABLE
1785 023710  112722  000006             MOVB    #RMDA,(R2)+
1786 023714  112722  000032             MOVB    #RMDF,(R2)+
1787 023720  112722  000034             MOVB    #RMDC,(R2)+
1788 023724  112722  000004             MOVB    #RMBA,(R2)+
1789 023730  112722  000002             MOVB    #RMWC,(R2)+
1790 023734  112722  000000             MOVB    #RMCS1,(R2)+
1791 023740  112712  000200             MOVB    #200,(R2)
1792
1793 023744  004737  040360             JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
023750  000404                          BR      31$             ;GO TO 31$ IF NO ERROR
023752  000240                          NOP                     ;RETURN HERE IF ERROR
023754  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
023756  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1794 023762                     31$:
1795
1796                            ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
023762  004737  040024             JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1797
1798                            ;WAIT FOR COMMAND TO COMPLETE
023766  004737  040732             JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1799
1800                            ;GO READ STATUS FOR READ OPERATION
1801 023772  004737  040110             JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
```

T14 WRITE, READ W/ HCE ERROR

```
        023776  000404                          BR      32$             ;GO TO 32$ IF NO ERROR
        024000  000240                          NOP                     ;RETURN HERE IF ERROR
        024002  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        024004  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1802    024010                          32$:
1803
1804                                    ;CHECK FOR ERRORS DURING READ OPERATION
1805    024010  004737  041126                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        024014  000405                          BR      33$             ;GO TO 33$ IF NO ERROR
        024016  000240                          NOP                     ;RETURN HERE IF ERROR
        024020  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        024022  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024024  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1806    024030                          33$:
1807    024030  005704                          TST     R4              ;IS THIS THE FIRST HEADER WORD ?
1808    024032  001011                          BNE     34$             ;NO !!
1809
1810    024034  032703  020000                  BIT     #SSF,R3         ;SHOULD SSE BE SET ?
1811    024040  001042                          BNE     37$             ;YES !!
1812    024042  032703  010000                  BIT     #FMT16,R3       ;SHOULD FER BE SET ?
1813    024046  001067                          BNE     39$             ;YES !!
1814    024050  032703  140000                  BIT     #MSE!USE,R3     ;SHOULD BSE BE SET ?
1815    024054  001111                          BNE     41$             ;YES !!
1816
1817                                    ;VERIFY THAT A HEADER COMPARE ERROR WAS DETECTED
1818    024056                          34$:
1819    024056  032737  000200  001352          BIT     #HCE,RMER1I     ;WAS HCE DETECTED ??
1820    024064  001132                          BNE     43$             ;YES !!
1821
1822    024066  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        024072  000405                          BR      35$             ;GO TO 35$ IF NO ERROR
        024074  000240                          NOP                     ;RETURN HERE IF ERROR
        024076  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        024100  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024102  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1823    024106                          35$:
1824    024106  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1825    024114  012737  000200  001140          MOV     #HCE,$GDDAT     ;EXPECTED STATUS
1826    024122  010437  001174                  MOV     R4,$TMP0        ;R4 = HEADER WORD NUMBER AND
1827    024126  001002                          BNE     36$             ;ADJUST NUMBER.
1828    024130  005237  001174                  INC     $TMP0
1829    024134  010337  001176          36$:    MOV     R3,$TMP1        ;R3 = BIT POSITION
1830    024140  104344                          EMT     344
1831    024142  000137  024416                  JMP     46$
1832
1833                                    ;VERIFY THAT SSE BIT IS SET
1834    024146                          37$:
1835    024146  032737  000040  001400          BIT     #SSE,RMER2I     ;IS SSE SET ?
1836    024154  001076                          BNE     43$             ;YES !!
1837
1838    024156  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        024162  000405                          BR      38$             ;GO TO 38$ IF NO ERROR
        024164  000240                          NOP                     ;RETURN HERE IF ERROR
        024166  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        024170  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        024172  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1839    024176                          38$:
```

```
4       WRITE, READ W/ HCE ERROR

1840 024176  013737  001400  001142          MOV     RMER2I,$BDDAT   ;GET RECIEVED STATUS
1841 024204  013737  001400  001140          MOV     RMER2I,$GDDAT   ;GET EXPECTED STATUS AND
1842 024212  052737  000040  001140          BIS     #SSE,$GDDAT     ;SET SSE.
1843 024220  104224                          EMT     224
1844 024222  000137  024416                  JMP     46$
1845
1846                                 ;VERIFY THAT A FORMAT ERROR WAS DETECTED
1847 024226                          39$:
1848 024226  032737  000020  001352          BIT     #FER,RMER1I     ;WAS FER DETECTED ??
1849 024234  001046                          BNE     43$             ;YES !!
1850
1851 024236  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024242  000405                          BR      40$             ;GO TO 40$ IF NO ERROR
     024244  000240                          NOP                     ;RETURN HERE IF ERROR
     024246  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024250  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024252  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1852 024256                          40$:
1853 024256  012737  000020  001140          MOV     #FER,$GDDAT     ;EXPECTED STATUS
1854 024264  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
1855 024272  104343                          EMT     343
1856 024274  000137  024416                  JMP     46$
1857
1858                                 ;VERIFY THAT A BAD SECTOR ERROR WAS DETECTED
1859 024300                          41$:
1860 024300  032737  100000  001400          BIT     #BSE,RMER2I     ;WAS BSE DETECTED ??
1861 024306  001021                          BNE     43$             ;YES !!
1862
1863 024310  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024314  000405                          BR      42$             ;GO TO 42$ IF NO ERROR
     024316  000240                          NOP                     ;RETURN HERE IF ERROR
     024320  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024322  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024324  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1864 024330                          42$:
1865 024330  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
1866 024336  012737  100000  001140          MOV     #BSE,$GDDAT     ;EXPECTED STAIUS
1867 024344  104345                          EMT     345
1868 024346  000137  000350                  JMP     350
1869
1870                                 ;CHECK FOR OTHER ERRORS
1871 024352                          43$:
1872 024352  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     024356  000405                          BR      44$             ;GO TO 44$ IF NO ERROR
     024360  000240                          NOP                     ;RETURN HERE IF ERROR
     024362  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     024364  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     024366  000137  024416                  JMP     46$             ;GO TO 46$ IF ERROR
1873 024372                          44$:
1874 024372  006303                          ASL     R3              ;SHIFT TO NEXT BIT
1875 024374  001006                          BNE     45$             ;REPEAT IF NOT DONE
1876
1877 024376  005704                          TST     R4              ;SECOND HEADER WORD DONE??
1878 024400  001006                          BNE     46$             ;YES!!
1879 024402  012704  000002                  MOV     #2,R4           ;SETUP FOR 2ND HEADER WORD
1880 024406  012703  000001                  MOV     #1,R3           ;SETUP FOR HCE
1881 024412                          45$:
```

```
1882  024412  000137  022604                    JMP     2$
1883  024416                          46$:
1884
1885  024416                          47$:
1886
1887                                  ;;*****************************************************************
                                      ;*TEST 15        WRITE, READ W/ HCI & SSEI
                                      ;;*****************************************************************
      024416                          TST15:
      024416  000004                          SCOPE                   ;SCOPE CALL
      024420  000240                          NOP                     ;START OF TEST
      024422  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      024426  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      024432  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      024436  012737  000015  001226          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1888
1889                                  ;;*****************************************************************
1890                                  ;LOOP #1        FORMAT,WRITE,READ
1891
1892  024444  012704  000000          1$:     MOV     #0,R4           ;HEADER WORD
1893  024450  012703  000001                  MOV     #1,R3           ;HCE BIT
1894  024454                          2$:
1895
1896                                  ;PREPARE THE DEVICE FOR FORMAT OPERATION
1897  024454  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      024460  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
      024462  000404                  BR      3$              ;GO TO 3$ IF NO ERROR
      024464  000240                  NOP                     ;RETURN HERE IF ERROR
      024466  104000                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      024470  000137  025720          JMP     38$             ;GO TO 38$ IF ERROR
1898
1899                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
1900  024474                          3$:
1901  024474  012737  001057  001446          MOV     #559.,RMDC0     ;CYLINDER = 559.
1902  024502  012737  001000  001420          MOV     #TA2,RMDA0      ;TRACK = 2, SECTOR = 0
1903  024510  012737  104254  001416          MOV     #BUFONE,RMBA0   ;BUS ADDRESS
1904  024516  012737  177376  001414          MOV     #-258.,RMWC0    ;2 + 256. WORDS (2' COMP)
1905  024524  012737  010000  001444          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
1906  024532  012737  000063  001412          MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA COMMAND
1907
1908                                  ;VERIFY THAT SECTOR IS NOT BAD
      024540  004737  034762                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
      024544  000405                  BR      4$              ;GO TO 4$ IF NO ERROR
      024546  104401  066006          TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
      024552  104000                  EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
      024554  000137  025720          JMP     37$             ;GO TO 37$ IF ERROR
1909  024560                          4$:
1910  024560  012737  067414  001174          MOV     #ONES,$TMP0     ;STARTING ADDRESS OF PATTERN
1911  024566  012737  000001  001176          MOV     #1,$TMP1        ;RANGE OF PATTERN
1912  024574  004737  037214                  JSR     PC,GENBUF       ;GO GENERATE BUFFER FOR FORMAT
```

```
1914
1915 024600  030364  104254                ;CHANGE HEADER WORD TO FORCE ERROR DURING WRITE
1916 024604  001403                        BIT     R3,BUFONE(R4)   ;SET OR RESET BIT??
1917 024606  040364  104254                BEQ     5$
1918 024612  000402                        BIC     R3,BUFONE(R4)   ;RESET BIT
1919 024614  050364  104254                BR      6$
1920                                       BIS     R3,BUFONE(R4)   ;SET BIT
1921
1922 024620                                ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
1923 024620  012702  001555                6$:
1924 024624  112722  000034                        MOV     #PUTINX,R2      ;R2 = BYTE ENTRY POSITION
1925 024630  112722  000006                        MOVB    #RMDC,(R2)+
1926 024634  112722  000004                        MOVB    #RMDA,(R2)+
1927 024640  112722  000002                        MOVB    #RMBA,(R2)+
1928 024644  112722  000032                        MOVB    #RMWC,(R2)+
1929 024650  112722  000000                        MOVB    #RMOF,(R2)+
1930 024654  112712  000200                        MOVB    #RMCS1,(R2)+
1931 024660                                        MOVB    #200,(R2)       ;TERMINATE TABLE
1932                                       7$:
1933
1934 024660  004737  040360                ;FORMAT THE DRIVE
     024664  000404                              JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     024666  000240                              BR      8$              ;GO TO 8$ IF NO ERROR
     024670  104000                              NOP                     ;RETURN HERE IF ERROR
     024672  000137  025720                      EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
1935 024676                                      JMP     37$             ;GO TO 37$ IF ERROR
1936                                       8$:
1937
     024676  004737  040024                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
1938                                             JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1939
     024702  004737  040732                ;WAIT FOR COMMAND TO COMPLETE
1940                                             JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1941
1942 024706  004737  040110                ;GO READ STATUS FOR FORMAT OPERATION
     024712  000404                              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     024714  000240                              BR      9$              ;GO TO 9$ IF NO ERROR
     024716  104000                              NOP                     ;RETURN HERE IF ERROR
     024720  000137  025720                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
1943 024724                                      JMP     37$             ;GO TO 37$ IF ERROR
1944                                       9$:
1945
1946 024724  004737  054166                ;VERIFY NO ERRORS DURING FORMAT
     024730  000405                              JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     024732  000240                              BR      10$             ;GO TO 10$ IF NO ERROR
     024734  104000                              NOP                     ;RETURN HERE IF ERROR
     024736  004736                              EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     024740  000137  025720                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1947 024744                                      JMP     37$             ;GO TO 37$ IF ERROR
1948                                       10$:
1949
1950 024744  012737  024754  001124        ;MOVE LOOP ADDRESSES TO NEXT OPERATION
1951 024752  000410                              MOV     #11$,$LPERR
1952                                             BR      12$             ;SKIP TO WRITE OPERATION
1953
1954                                       ;;*****************************************************************
                                           ;LOOP #2        WRITE,READ
```

```
                                        11$:
                                        ;PREPARE DEVICE FOR WRITE OPERATION
024754  004737  034032                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
024760  154130                                  .WOFD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
024762  000404                                  BR      12$             ;GO TO 12$ IF NO ERROR
024764  000240                                  NOP                     ;RETURN HERE IF ERROR
024766  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
024770  000137  025720                          JMP     37$             ;GO TO 37$ IF ERROR
024774                                  12$:
                                        ;WRITE DATA TO THE DRIVE
024774                                  13$:
024774  012737  013000  001444                  MOV     #FMT16!HCI!SSEI,RMOFO   ;INHIBIT HEADER COMPARE AND SKIP
                                                                        ;SKIP SECTOR ERRORS
025002  012737  000061  001412                  MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
025010  012737  104260  001416                  MOV     #BUFONE+4,RMBAO         ;LOAD STARTING BUFFER ADDRESS
025016  012702  001555                          MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
025022  112722  000006                          MOVB    #RMDA,(R2)+
025026  112722  000034                          MOVB    #RMDC,(R2)+
025032  112722  000032                          MOVB    #RMOF,(R2)+
025036  112722  000004                          MOVB    #RMBA,(R2)+
025042  112722  000002                          MOVB    #RMWC,(R2)+
025046  112722  000000                          MOVB    #RMCS1,(R2)+
025052  112722  000200                          MOVB    #200,(R2)+              ;TERMINATE TABLE

025056  004737  040360                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
025062  000404                                  BR      14$             ;GO TO 14$ IF NO ERROR
025064  000240                                  NOP                     ;RETURN HERE IF ERROR
025066  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
025070  000137  025720                          JMP     37$             ;GO TO 37$ IF ERROR
025074                                  14$:
                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
025074  004737  040024                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
025100  004737  040732                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE

                                        ;GO READ STATUS FOR WRITE COMMAND
025104  004737  040110                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
025110  000404                                  BR      15$             ;GO TO 15$ IF NO ERROR
025112  000240                                  NOP                     ;RETURN HERE IF ERROR
025114  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
025116  000137  025720                          JMP     37$             ;GO TO 37$ IF ERROR
025122                                  15$:
                                        ;CHECK FOR ERRORS DURING WRITE OPERATION
025122  004737  041126                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
```

```
        025126  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        025130  000240                          NOP                     ;RETURN HERE IF ERROR
        025132  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        025134  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        025136  000137  025720                  JMP     37$             ;GO TO 37$ IF ERROR
1990    025142  005037  001140          16$:    CLR     $GDDAT                  ;EXPECTED STATUS
1991
1992    025146  032737  000220  001352          BIT     #HCE!FER,RMER1I         ;ANY ERROR??
1993    025154  001407                          BEQ     17$
1994    025156  013737  001352  001142          MOV     RMER1I,$BDDAT           ;RECEIVED STATUS
1995    025164  042737  177557  001142          BIC     #^C<HCE!FER>,$BDDAT     ;CLEAR DONT CARES
1996    025172  000412                          BR      18$
1997    025174                          17$:
1998    025174  013737  001400  001142          MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
1999    025202  032737  100000  001400          BIT     #BSE,RMER2I             ;ANY BAD SECTOR ERROR ?
2000    025210  001406                          BEQ     19$                     ;NO !!
2001    025212  042737  077777  001142          BIC     #^CBSE,$BDDAT           ;SAVE BSE FOR ERROR
2002    025220                          18$:
2003    025220  104346                          EMT     346
2004    025222  000137  025720                  JMP     37$
2005    025226                          19$:
2006    025226  032737  000040  001400          BIT     #SSE,RMER2I             ;ANY SKIP SECTOR ERROR ?
2007    025234  001406                          BEQ     20$
2008    025236  043737  177737  001142          BIC     ^CSSE,$BDDAT            ;SAVE SSE FOR ERROR
2009    025244  104227                          EMT     227
2010    025246  000137  025720                  JMP     37$
2011    025252                          20$:
2012    025252  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        025256  000405                          BR      21$             ;GO TO 21$ IF NO ERROR
        025260  000240                          NOP                     ;RETURN HERE IF ERROR
        025262  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        025264  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        025266  000137  025720                  JMP     37$             ;GO TO 37$ IF ERROR
2013    025272                          21$:
2014    025272  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        025276  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
        025300  000240                          NOP                     ;RETURN HERE IF ERROR
        025302  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        025304  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        025306  000137  025720                  JMP     37$             ;GO TO 37$ IF ERROR
2015    025312                          22$:
2016
2017                                    ;CHANGE LOOP ADDRESSES
2018    025312  012737  025322  001124          MOV     #23$,$LPERR
2019    025320  000410                          BR      24$
2020
2021                                    ;;*****************************************************************
2022                                    ;LOOP #3        READ
2023
2024    025322                          23$:
2025
2026                                    ;PREPARE DEVICE FOR READ OPERATION
2027    025322  004737  034032                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        025326  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
 025330  000404                        BR      24$             ;GO TO 24$ IF NO ERROR
 025332  000240                        NOP                     ;RETURN HERE IF ERROR
 025334  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
 025336  000137  025720                JMP     37$             ;GO TO 37$ IF ERROR
2028 025342                     24$:
2029
2030                            ;READ DATA FROM DEVICE
2031 025342                     25$:
2032 025342  012737  000071  001412    MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
2033 025350  012737  105264  001416    MOV     #BUFTWO+4,RMBA0         ;LOAD STARTING BUFFER ADDRESS
2034 025356  012702  001555            MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2035 025362  112722  000006            MOVB    #RMDA,(R2)+
2036 025366  112722  000032            MOVB    #RMOF,(R2)+
2037 025372  112722  000034            MOVB    #RMDC,(R2)+
2038 025376  112722  000004            MOVB    #RMBA,(R2)+
2039 025402  112722  000002            MOVB    #RMWC,(R2)+
2040 025406  112722  000000            MOVB    #RMCS1,(R2)+
2041 025412  112712  000200            MOVB    #200,(R2)
2042
2043 025416  004737  040360            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
 025422  000404                        BR      26$             ;GO TO 26$ IF NO ERROR
 025424  000240                        NOP                     ;RETURN HERE IF ERROR
 025426  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
 025430  000137  025720                JMP     37$             ;GO TO 37$ IF ERROR
2044 025434                     26$:
2045
2046                            ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
 025434  004737  040024            JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2047
2048                            ;WAIT FOR COMMAND TO COMPLETE
 025440  004737  040732            JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2049
2050                            ;GO READ STATUS FOR READ OPERATION
2051 025444  004737  040110            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
 025450  000404                        BR      27$             ;GO TO 27$ IF NO ERROR
 025452  000240                        NOP                     ;RETURN HERE IF ERROR
 025454  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
 025456  000137  025720                JMP     37$             ;GO TO 37$ IF ERROR
2052 025462                     27$:
2053
2054                            ;CHECK FOR ERRORS DURING READ OPERATION
2055 025462  004737  041126            JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
 025466  000405                        BR      28$             ;GO TO 28$ IF NO ERROR
 025470  000240                        NOP                     ;RETURN HERE IF ERROR
 025472  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
 025474  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 025476  000137  025720                JMP     37$             ;GO TO 37$ IF ERROR
2056 025502  005037  001140    28$:    CLR     $GDDAT                  ;EXPECTED STATUS
2057
2058 025506  032737  000220  001352    BIT     #HCE!FER,RMER1I         ;ANY ERROR ?
2059 025514  001407                    BEQ     29$                     ;NO!!
2060 025516  013737  001352  001142    MOV     RMER1I,$BDDAT           ;RECEIVED STATUS
2061 025524  042737  177557  001142    BIC     #^C<HCE!FER>,$BDDAT     ;CLEAR DONT CARES
```

```
2062 025552 000412                          BR      30$
2063 025534                          29$:
2064 025534 013737 001400 001142            MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
2065 025542 032737 100000 001400            BIT     #BSE,RMER2I             ;ANY BAD SECTOR ERROR ?
2066 025550 001406                          BEQ     31$                     ;NO !!
2067 025552 042737 077777 001142            BIC     #^CBSE,$BDDAT           ;CLEAR DONT CARES
2068 025560                          30$:
2069 025560 104346                          EMT     346
2070 025562 000137 025720                   JMP     37$
2071 025566                          31$:
2072 025566 032737 000040 001400            BIT     #SSE,RMER2I             ;ANY SKIP SECTOR ERROR ?
2073 025574 001406                          BEQ     32$                     ;NO !!
2074 025576 043737 177737 001142            BIC     ^CSSE,$BDDAT            ;SAVE SSE FOR ERROR
2075 025604 104227                          EMT     227
2076 025606 000137 025720                   JMP     37$
2077 025612                          32$:
2078 025612 004737 054166                   JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     025616 000405                          BR      33$             ;GO TO 33$ IF NO ERROR
     025620 000240                          NOP                     ;RETURN HERE IF ERROR
     025622 104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     025624 004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025626 000137 025720                   JMP     37$             ;GO TO 37$ IF ERROR
2079 025632                          33$:
2080 025632 004737 041760                   JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     025636 000405                          BR      34$             ;GO TO 34$ IF NO ERROR
     025640 000240                          NOP                     ;RETURN HERE IF ERROR
     025642 104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     025644 004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     025646 000137 025720                   JMP     37$             ;GO TO 37$ IF ERROR
2081 025652                          34$:
2082 025652 004737 037462                   JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
     025656 104260                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
     025660 105264                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
     025662 000404                          BR      35$             ;GO TO 35$ IF NO ERROR
     025664 000240                          NOP                     ;RETURN HERE IF ERROR
     025666 104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
     025670 000137 025720                   JMP     37$             ;GO TO 37$ IF ERROR
2083 025674                          35$:
2084
2085 025674 006303                          ASL     R3                      ;SHIFT HCE BIT
2086 025676 001006                          BNE     36$                     ;CONTINUE IF NOT DONE
2087 025700 005704                          TST     R4                      ;SECOND HEADER DONE??
2088 025702 001006                          BNE     37$                     ;YES!!
2089 025704 012703 000001                   MOV     #1,R3                   ;START WITH BIT 0
2090 025710 012704 000002                   MOV     #2,R4                   ;DO SECOND HEADER WORD
2091 025714                          36$:
2092 025714 000137 024454                   JMP     2$
2093 025720                          37$:
2094
2095 025720                          38$:
2096
2097                                 ;;*****************************************************************
                                     ;*TEST 16       WRITE, READ W/ IVC ERROR
                                     ;;*****************************************************************
     025720                          TST16:
     025720 000004                          SCOPE                   ;SCOPE CALL
     025722 000240                          NOP                     ;START OF TEST
```

```
        025724  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        025730  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        025734  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        025740  012737  000016  001226          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  2098
  2099                                  ;;*****************************************************************
  2100                                  ;LOOP #1        WRITE,READ
  2101
  2102                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  2103 025746                           1$:
  2104 025746   012737  001057  001446          MOV     #559.,RMDC0     ;CYLINDER = 559.
  2105 025754   012737  001000  001420          MOV     #TA2,RMDA0      ;TRACK = 2, SECTOR = 0
  2106 025762   012737  104254  001416          MOV     #BUFONE,RMBA0   ;BUS ADDRESS
  2107 025770   012737  177400  001414          MOV     #-256.,RMWC0    ;256. WORDS (2'S COMP)
  2108 025776   012737  010000  001444          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
  2109 026004   012737  000060  001412          MOV     #WD,RMCS10      ;WRITE HEADER AND DATA COMMAND
  2110
  2111                                  ;VERIFY THAT SECTOR IS NOT BAD
        026012  004737  034762                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        026016  000405                          BR      2$              ;GO TO 2$ IF NO ERROR
        026020  104401  066006                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        026024  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        026026  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
  2112 026032                           2$:
  2113 026032   012737  001420  001174          MOV     #RMDA0,$TMP0    ;USE SECTOR FOR DATA PATTERN
  2114 026040   012737  000001  001176          MOV     #1,$TMP1
  2115 026046   004737  037214                  JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
  2116
  2117                                  ;PREPARE DEVICE FOR WRITE OPERATION
  2118 026052   004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        026056  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        026060  000404                          BR      3$              ;GO TO 3$ IF NO ERROR
        026062  000240                          NOP                     ;RETURN HERE IF ERROR
        026064  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        026066  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
  2119 026072                           3$:
  2120
  2121                                  ;RESET VOLUME VALID
  2122 026072   112737  000024  001555          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
        026100  112737  000200  001556          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        026106  012737  000001  001436          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
        026114  004737  040360                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
        026120  000404                          BR      4$              ;GO TO 4$ IF NO ERROR
        026122  000240                          NOP                     ;RETURN HERE IF ERROR
        026124  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        026126  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
  2123 026132                           4$:
  2124
  2125                                  ;SETUP PARAMETERS AND EXCUTE WRITE DATA COMMAND
  2126 026132                           5$:
```

```
 2127 026132  012737  000061  001412          MOV     #WD!GO,RMCS1O           ;WRITE DATA COMMAND
 2128 026140  012737  000000  001436          MOV     #0,RMMR1O               ;RESET DIAGNOSTIC MODE
 2129 026146  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2130 026152  112722  000024                  MOVB    #RMMR1,(R2)+
 2131 026156  112722  000006                  MOVB    #RMDA,(R2)+
 2132 026162  112722  000034                  MOVB    #RMDC,(R2)+
 2133 026166  112722  000032                  MOVB    #RMOF,(R2)+
 2134 026172  112722  000002                  MOVB    #RMWC,(R2)+
 2135 026176  112722  000004                  MOVB    #RMBA,(R2)+
 2136 026202  112722  000000                  MOVB    #RMCS1,(R2)+
 2137 026206  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 2138
 2139 026212  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      026216  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
      026220  000240                          NOP                     ;RETURN HERE IF ERROR
      026222  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      026224  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
 2140 026230                          6$:
 2141
 2142                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      026230  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2143
 2144                                 ;WAIT FOR COMMAND TO COMPLETE
      026234  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2145
 2146                                 ;GO READ STATUS FOR WRITE COMMAND
 2147 026240  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      026244  000404                          BR      7$              ;GO TO 7$ IF NO ERROR
      026246  000240                          NOP                     ;RETURN HERE IF ERROR
      026250  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      026252  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
 2148 026256                          7$:
 2149
 2150                                 ;CHECK FOR ERRORS DURING WRITE OPERATION
 2151 026256  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      026262  000405                          BR      8$              ;GO TO 8$ IF NO ERROR
      026264  000240                          NOP                     ;RETURN HERE IF ERROR
      026266  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      026270  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026272  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
 2152 026276                          8$:
 2153 026276  032737  010000  001400          BIT     #IVC,RMER2I             ;WAS "IVC" DETECTED??
 2154 026304  001024                          BNE     10$                     ;YES!!
 2155
 2156 026306  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      026312  000405                          BR      9$              ;GO TO 9$ IF NO ERROR
      026314  000240                          NOP                     ;RETURN HERE IF ERROR
      026316  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      026320  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026322  000137  026740                  JMP     22$             ;GO TO 22$ IF ERROR
 2157 026326                          9$:
 2158 026326  013737  001400  001142          MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
 2159 026334  013737  001400  001140          MOV     RMER2I,$GDDAT           ;EXPECTED STATUS
 2160 026342  052737  010000  001140          BIS     #IVC,$GDDAT
 2161 026350  104342                          EMT     342
 2162 026352  000137  026740                  JMP     22$
 2163 026356                          10$:
```

```
 2164 026356  004737  041760                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      026362  000405                        BR      11$             ;GO TO 11$ IF NO ERROR
      026364  000240                        NOP                     ;RETURN HERE IF ERROR
      026366  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      026370  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      026372  000137  026740                JMP     22$             ;GO TO 22$ IF ERROR
 2165 026376                          11$:
 2166
 2167                                 ;CHANGE LOOP ADDRESSES
 2168 026376  012737  026406  001124        MOV     #12$,$LPERR
 2169 026404  000410                        BR      13$
 2170
 2171                                 ;;*****************************************************************
 2172                                 ;LOOP #2        READ
 2173
 2174 026406                          12$:
 2175
 2176                                 ;PREPARE DEVICE FOR READ OPERATION
 2177 026406  004737  034032                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      026412  154130                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
      026414  000404                        BR      13$             ;GO TO 13$ IF NO ERROR
      026416  000240                        NOP                     ;RETURN HERE IF ERROR
      026420  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      026422  000137  026740                JMP     22$             ;GO TO 22$ IF ERROR
 2178 026426                          13$:
 2179
 2180                                 ;RESET VOLUME VALID
 2181 026426  112737  000024  001555        MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
      026434  112737  000200  001556        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      026442  012737  000001  001436        MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
      026450  004737  040360                JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
      026454  000404                        BR      14$             ;GO TO 14$ IF NO ERROR
      026456  000240                        NOP                     ,RETURN HERE IF ERROR
      026460  104000                        EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      026462  000137  026740                JMP     22$             ;GO TO 22$ IF ERROR
 2182 026466                          14$:
 2183
 2184                                 ;READ DATA FROM DEVICE
 2185 026466                          15$:
 2186 026466  012737  000071  001412        MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
 2187 026474  012737  105260  001416        MOV     #BUFTWO,RMBA0           ;LOAD STARTING BUFFER ADDRESS
 2188 026502  012737  000000  001436        MOV     #0,RMMR10               ;RESET DIANOSTIC MODE
 2189 026510  012702  001555                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2190 026514  112722  000024                MOVB    #RMMR1,(R2)+
 2191 026520  112722  000006                MOVB    #RMDA,(R2)+
 2192 026524  112722  000032                MOVB    #RMOF,(R2)+
 2193 026530  112722  000034                MOVB    #RMDC,(R2)+
 2194 026534  112722  000004                MOVB    #RMBA,(R2)+
 2195 026540  112722  000002                MOVB    #RMWC,(R2)+
 2196 026544  112722  000000                MOVB    #RMCS1,(R2)+
```

```
2197 026550  112712  000200                    MOVB    #200,(R2)
2198
2199 026554  004737  040360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     026560  000404                            BR      16$             ;GO TO 16$ IF NO ERROR
     026562  000240                            NOP                     ;RETURN HERE IF ERROR
     026564  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     026566  000137  026740                    JMP     22$             ;GO TO 22$ IF ERROR
2200 026572                            16$:
2201
2202                                   ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     026572  004737  040024                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2203
2204                                   ;WAIT FOR COMMAND TO COMPLETE
     026576  004737  040732                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2205
2206                                   ;GO READ STATUS FOR READ OPERATION
2207 026602  004737  040110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     026606  000404                            BR      17$             ;GO TO 17$ IF NO ERROR
     026610  000240                            NOP                     ;RETURN HERE IF ERROR
     026612  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     026614  000137  026740                    JMP     22$             ;GO TO 22$ IF ERROR
2208 026620                            17$:
2209
2210                                   ;CHECK FOR ERRORS DURING READ OPERATION
2211 026620  004737  041126                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     026624  000405                            BR      18$             ;GO TO 18$ IF NO ERROR
     026626  000240                            NOP                     ;RETURN HERE IF ERROR
     026630  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     026632  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026634  000137  026740                    JMP     22$             ;GO TO 22$ IF ERROR
2212 026640                            18$:
2213 026640  032737  010000  001400            BIT     #IVC,RMER2I             ;WAS "IVC" DETECTED??
2214 026646  001024                            BNE     20$                     ;YES!!
2215
2216 026650  004737  054166                    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     026654  000405                            BR      19$             ;GO TO 19$ IF NO ERROR
     026656  000240                            NOP                     ;RETURN HERE IF ERROR
     026660  104000                            EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     026662  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026664  000137  026740                    JMP     22$             ;GO TO 22$ IF ERROR
2217 026670                            19$:
2218 026670  013737  001400  001140            MOV     RMER2I,$GDDAT           ;EXPECTED STATUS
2219 026676  052737  010000  001140            BIS     #IVC,$GDDAT
2220 026704  013737  001400  001142            MOV     RMER2I,$BDDAT           ;RECEIVED STATUS
2221 026712  104342                            EMT     342
2222 026714  000137  026740                    JMP     22$
2223 026720                            20$:
2224 026720  004737  041760                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     026724  000405                            BR      21$             ;GO TO 21$ IF NO ERROR
     026726  000240                            NOP                     ;RETURN HERE IF ERROR
     026730  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     026732  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     026734  000137  026740                    JMP     22$             ;GO TO 22$ IF ERROR
2225 026740                            21$:
2226
2227 026740                            22$:
2228
```

```
2229                                    ;;*****************************************************************
                                        ;*TEST 17        WRITE, READ W/ ABORT
                                        ;;*****************************************************************
     026740                             TST17:
     026740  000004                             SCOPE                   ;SCOPE CALL
     026742  000240                             NOP                     ;START OF TEST
     026744  012706  001100                     MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     026750  013700  001276                     MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     026754  013701  001466                     MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     026760  012737  000017  001226             MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2230
2231                                    ;;*****************************************************************
2232                                    ;LOOP #1        WRITE,READ
2233
2234                                    ;LOAD PARAMETERS AND GENERATE DATA BUFFER
2235 026766  012737  001057  001446             MOV     #559.,RMDC0             ;CYLINDER = 559.
2236 026774  012737  001000  001420             MOV     #TA2,RMDA0              ;TRACK = 2, SECTOR = 0
2237 027002  012737  104254  001416             MOV     #BUFONE,RMBA0           ;BUS ADDRESS
2238 027010  012737  177400  001414             MOV     #-256.,RMWC0            ;256. WORDS (2'S COMP)
2239 027016  012737  010000  001444             MOV     #FMT16,RMOF0            ;16 BIT FORMAT
2240 027024  012737  000061  001412             MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2241 027032                             1$:
2242
2243                                    ;PREPARE DEVICE FOR WRITE OPERATION
2244 027032  004737  034032                     JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     027036  154130                             .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
     027040  000404                             BR      2$              ;GO TO 2$ IF NO ERROR
     027042  000240                             NOP                     ;RETURN HERE IF ERROR
     027044  104000                             EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     027046  000137  027656                     JMP     21$             ;GO TO 21$ IF ERROR
2245 027052                             2$:
2246
2247                                    ;SET UNSAFE ERROR
2248 027052  112737  000014  001555             MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
     027060  112737  000200  001556             MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     027066  012737  040000  001426             MOV     #UNS,RMER10     ;SET RMER1 OUTPUT BUFFER = UNS
     027074  004737  040360                     JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
     027100  000404                             BR      3$              ;GO TO 3$ IF NO ERROR
     027102  000240                             NOP                     ;RETURN HERE IF ERROR
     027104  104000                             EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     027106  000137  027656                     JMP     21$             ;GO TO 21$ IF ERROR
2249 027112                             3$:
2250
2251                                    ;WRITE DATA TO THE DRIVE
2252 027112                             4$:
2253 027112  012737  000061  001412             MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
2254 027120  012702  001555                     MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2255 027124  112722  000006                     MOVB    #RMDA,(R2)+
2256 027130  112722  000034                     MOVB    #RMDC,(R2)+
2257 027134  112722  000032                     MOVB    #RMOF,(R2)+
```

```
2258 027140  112722  000004                    MOVB    #RMBA,(R2)+
2259 027144  112722  000002                    MOVB    #RMWC,(R2)+
2260 027150  112722  000000                    MOVB    #RMCS1,(R2)+
2261 027154  112722  000200                    MOVB    #200,(R2)+              ;TERMINATE TABLE
2262
2263 027160  004737  040360                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027164  000404                            BR      5$              ;GO TO 5$ IF NO ERROR
     027166  000240                            NOP                     ;RETURN HERE IF ERROR
     027170  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     027172  000137  027656                    JMP     21$             ;GO TO 21$ IF ERROR
2264 027176                            5$:
2265
2266                                   ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     027176  004737  040024                    JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2267
2268                                   ;GO GET REGISTER STATUS
2269 027202  004737  040110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027206  000404                            BR      6$              ;GO TO 6$ IF NO ERROR
     027210  000240                            NOP                     ;RETURN HERE IF ERROR
     027212  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027214  000137  027656                    JMP     21$             ;GO TO 21$ IF ERROR
2270 027220                            6$:
2271 027220  032737  020000  001350            BIT     #PIP,RMDSI              ;WAS COMMAND EXECUTED?
2272 027226  001412                            BEQ     7$              ;NO, GO WAIT
2273 027230  013737  001350  001140            MOV     RMDSI,$GDDAT            ;EXPECTED STATUS
2274 027236  042737  020000  001140            BIC     #PIP,$GDDAT
2275 027244  013737  001350  001142            MOV     RMDSI,$BDDAT            ;RECEIVED STATUS
2276 027252  104347                            EMT     347
2277 027254                            7$:
2278
2279                                   ;WAIT FOR COMMAND TO COMPLETE
     027254  004737  040732                    JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2280
2281                                   ;GO GET REGISTER STATUS
2282 027260  004737  040110                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027264  000404                            BR      8$              ;GO TO 8$ IF NO ERROR
     027266  000240                            NOP                     ;RETURN HERE IF ERROR
     027270  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027272  000137  027656                    JMP     21$             ;GO TO 21$ IF ERROR
2283 027276                            8$:
2284 027276  004737  041126                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     027302  000405                            BR      9$              ;GO TO 9$ IF NO ERROR
     027304  000240                            NOP                     ;RETURN HERE IF ERROR
     027306  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     027310  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027312  000137  027656                    JMP     21$             ;GO TO 21$ IF ERROR
2285 027316                            9$:
2286 027316  004737  041760                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     027322  000405                            BR      10$             ;GO TO 10$ IF NO ERROR
     027324  000240                            NOP                     ;RETURN HERE IF ERROR
     027326  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     027330  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027332  000137  027656                    JMP     21$             ;GO TO 21$ IF ERROR
2287 027336                            10$:
2288
2289                                   ;CHANGE LOOP ADDRESSES
2290 027336  012737  027346  001124            MOV     #11$,$LPERR
```

```
2291 027344  000410                        BR      12$
2292
2293                                ;;*****************************************************************
2294                                ;LOOP #2        READ
2295
2296 027346                         11$:
2297
2298                                ;PREPARE DEVICE FOR READ OPERATION
2299 027346  004737  034032                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     027352  154130                        .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
     027354  000404                        BR      12$             ;GO TO 12$ IF NO ERROR
     027356  000240                        NOP                     ;RETURN HERE IF ERROR
     027360  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     027362  000137  027656                JMP     21$             ;GO TO 21$ IF ERROR
2300 027366                         12$:
2301
2302                                ;SET UNSAFE ERROR
2303 027366  112737  000014  001555        MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
     027374  112737  000200  001556        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     027402  012737  040000  001426        MOV     #UNS,RMER10     ;SET RMER1 OUTPUT BUFFER = UNS
     027410  004737  040360                JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
     027414  000404                        BR      13$             ;GO TO 13$ IF NO ERROR
     027416  000240                        NOP                     ;RETURN HERE IF ERROR
     027420  104000                        EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     027422  000137  027656                JMP     21$             ;GO TO 21$ IF ERROR
2304 027426                         13$:
2305
2306                                ;READ DATA FROM DEVICE
2307 027426                         14$:
2308 027426  012737  000071  001412        MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
2309 027434  012702  001555                MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
2310 027440  112722  000006                MOVB    #RMDA,(R2)+
2311 027444  112722  000032                MOVB    #RMOF,(R2)+
2312 027450  112722  000034                MOVB    #RMDC,(R2)+
2313 027454  112722  000004                MOVB    #RMBA,(R2)+
2314 027460  112722  000002                MOVB    #RMWC,(R2)+
2315 027464  112722  000000                MOVB    #RMCS1,(R2)+
2316 027470  112712  000200                MOVB    #200,(R2)
2317
2318 027474  004737  040360                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027500  000404                        BR      15$             ;GO TO 15$ IF NO ERROR
     027502  000240                        NOP                     ;RETURN HERE IF ERROR
     027504  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     027506  000137  027656                JMP     21$             ;GO TO 21$ IF ERROR
2319 027512                         15$:
2320
2321                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     027512  004737  040024                JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2322
2323                                ;SEE IF DEVICE STARTED COMMAND
```

```
2324 027516  004737  040110            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027522  000404                    BR      16$             ;GO TO 16$ IF NO ERROR
     027524  000240                    NOP                     ;RETURN HERE IF ERROR
     027526  104000                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027530  000137  027656            JMP     21$             ;GO TO 21$ IF ERROR
2325 027534                        16$:
2326 027534  032737  020000  001350    BIT     #PIP,RMDSI              ;WAS COMMAND EXECUTED??
2327 027542  001414                    BEQ     17$                     ;NO!!
2328 027544  013737  001350  001140    MOV     RMDSI,$GDDAT            ;EXPECTED STATUS
2329 027552  042737  020000  001140    BIC     #PIP,$GDDAT
2330 027560  013737  001350  001142    MOV     RMDSI,$BDDAT            ;RECEIVED STATUS
2331 027566  104347                    EMT     347
2332 027570  000137  027656            JMP     21$
2333 027574                        17$:
2334
2335                               ;WAIT FOR COMMAND TO COMPLETE
     027574  004737  040732            JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2336
2337                               ;GO READ STATUS FOR READ OPERATION
2338 027600  004737  040110            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027604  000404                    BR      18$             ;GO TO 18$ IF NO ERROR
     027606  000240                    NOP                     ;RETURN HERE IF ERROR
     027610  104000                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027612  000137  027656            JMP     21$             ;GO TO 21$ IF ERROR
2339 027616                        18$:
2340 027616  004737  041126            JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     027622  000405                    BR      19$             ;GO TO 19$ IF NO ERROR
     027624  000240                    NOP                     ;RETURN HERE IF ERROR
     027626  104000                    EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     027630  004736                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027632  000137  027656            JMP     21$             ;GO TO 21$ IF ERROR
2341 027636                        19$:
2342 027636  004737  041760            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     027642  000405                    BR      20$             ;GO TO 20$ IF NO ERROR
     027644  000240                    NOP                     ;RETURN HERE IF ERROR
     027646  104000                    EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     027650  004736                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027652  000137  027656            JMP     21$             ;GO TO 21$ IF ERROR
2343 027656                        20$:
2344
2345 027656                        21$:
2346
2347                               ;;*****************************************************************
                                   ;*TEST 20        WRITE, READ EARLY PEAK SHIFT
                                   ;;*****************************************************************
     027656                        TST20:
     027656  000004                    SCOPE                   ;SCOPE CALL
     027660  000240                    NOP                     ;START OF TEST
     027662  012706  001100            MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     027666  013700  001276            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     027672  013701  001466            MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     027676  012737  000020  001226    MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2348
2349                               ;;*****************************************************************
2350                               ;LOOP #1        FORMAT,WRITE,READ
2351
2352 027704                        1$:
```

```
                                        ;PREPARE THE DEVICE FOR FORMAT OPERATION
027704  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
027710  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
027712  000404                              BR      2$              ;GO TO 2$ IF NO ERROR
027714  000240                              NOP                     ;RETURN HERE IF ERROR
027716  104000                              EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
027720  000137  030664                      JMP     25$             ;GO TO 25$ IF ERROR
027724                                  2$:

                                        ;LOAD PARAMETERS AND GENERATE DATA BUFFER
027724  012737  001057  001446              MOV     #559.,RMDCO             ;CYLINDER = 559.
027732  012737  001000  001420              MOV     #TA2,RMDAO              ;TRACK = 2, SECTOR = 0
027740  012737  104254  001416              MOV     #BUFONE,RMBAO           ;BUS ADDRESS
027746  012737  177376  001414              MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
027754  012737  010000  001444              MOV     #FMT16,RMOFO            ;16 BIT FORMAT
027762  012737  000063  001412              MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND

                                        ;VERIFY THAT SECTOR IS NOT BAD
027770  004737  034762                      JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
027774  000405                              BR      3$              ;GO TO 3$ IF NO ERROR
027776  104401  066006                      TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
030002  104000                              EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
030004  000137  030664                      JMP     25$             ;GO TO 25$ IF ERROR
030010                                  3$:
030010  012737  067670  001174              MOV     #EARLY,$TMP0            ;STARTING ADDRESS OF PATTERN
030016  012737  000001  001176              MOV     #1,$TMP1                ;RANGE OF PATTERN
030024  004737  037214                      JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT

                                        ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
030030  012702  001555                      MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
030034  112722  000034                      MOVB    #RMDC,(R2)+
030040  112722  000006                      MOVB    #RMDA,(R2)+
030044  112722  000004                      MOVB    #RMBA,(R2)+
030050  112722  000002                      MOVB    #RMWC,(R2)+
030054  112722  000032                      MOVB    #RMOF,(R2)+
030060  112722  000000                      MOVB    #RMCS1,(R2)+
030064  112712  000200                      MOVB    #200,(R2)               ;TERMINATE TABLE
030070                                  4$:
030070  004737  040360                      JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
030074  000404                              BR      5$              ;GO TO 5$ IF NO ERROR
030076  000240                              NOP                     ;RETURN HERE IF ERROR
030100  104000                              EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
030102  000137  030664                      JMP     25$             ;GO TO 25$ IF ERROR
030106                                  5$:

                                        ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
030106  004737  040024                      JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE

                                        ;WAIT FOR COMMAND TO COMPLETE
```

```
                030112  004737  040732                 JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2388
 2389                                           ;GO READ STATUS FOR FORMAT OPERATION
 2390           030116  004737  040110                 JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                030122  000404                         BR      6$              ;GO TO 6$ IF NO ERROR
                030124  000240                         NOP                     ;RETURN HERE IF ERROR
                030126  104000                         EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                030130  000137  030664                 JMP     25$             ;GO TO 25$ IF ERROR
 2391           030134                         6$:
 2392
 2393                                           ;VERIFY NO ERRORS DURING FORMAT
 2394           030134  004737  054166                 JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
                030140  000405                         BR      7$              ;GO TO 7$ IF NO ERROR
                030142  000240                         NOP                     ;RETURN HERE IF ERROR
                030144  104000                         EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
                030146  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                030150  000137  030664                 JMP     25$             ;GO TO 25$ IF ERROR
 2395           030154                         7$:
 2396
 2397                                           ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2398           030154  012737  030164  001124         MOV     #8$,$LPERR
 2399           030162  000410                         BR      9$
 2400
 2401                                           ;;*****************************************************************
 2402                                           ;LOOP #2        WRITE,READ
 2403
 2404           030164                         8$:
 2405
 2406                                           ;PREPARE DEVICE FOR WRITE OPERATION
 2407           030164  004737  034032                 JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                030170  154130                         .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                               ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                               ;VERIFY PACK ACKNOWLEDGE
                                                                               ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                               ;VERIFY RECALIBRATION
                030172  000404                         BR      9$              ;GO TO 9$ IF NO ERROR
                030174  000240                         NOP                     ;RETURN HERE IF ERROR
                030176  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                030200  000137  030664                 JMP     25$             ;GO TO 25$ IF ERROR
 2408           030204                         9$:
 2409
 2410                                           ;WRITE DATA TO THE DRIVE
 2411           030204                         10$:
 2412           030204  012737  177400  001414         MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
 2413           030212  012737  104260  001416         MOV     #BUFONE+4,RMBAO         ;CHANGE ADDRESS
 2414           030220  012737  000061  001412         MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2415           030226  012702  001555                 MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2416           030232  112722  000006                 MOVB    #RMDA,(R2)+
 2417           030236  112722  000034                 MOVB    #RMDC,(R2)+
 2418           030242  112722  000032                 MOVB    #RMOF,(R2)+
 2419           030246  112722  000004                 MOVB    #RMBA,(R2)+
 2420           030252  112722  000002                 MOVB    #RMWC,(R2)+
 2421           030256  112722  000000                 MOVB    #RMCS1,(R2)+
 2422           030262  112722  000200                 MOVB    #200,(R2)+              ;TERMINATE TABLE
```

```
                                                ;
030266  004737  040360                          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
030272  000404                                  BR      11$             ;GO TO 11$ IF NO ERROR
030274  000240                                  NOP                     ;RETURN HERE IF ERROR
030276  104000                                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
030300  000137  030664                          JMP     25$             ;GO TO 25$ IF ERROR
030304                                  11$:
                                                ;
                                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
030304  004737  040024                          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
                                                ;
                                                ;WAIT FOR COMMAND TO COMPLETE
030310  004737  040732                          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
                                                ;
                                                ;GO READ STATUS FOR WRITE COMMAND
030314  004737  040110                          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
030320  000404                                  BR      12$             ;GO TO 12$ IF NO ERROR
030322  000240                                  NOP                     ;RETURN HERE IF ERROR
030324  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
030326  000137  030664                          JMP     25$             ;GO TO 25$ IF ERROR
030332                                  12$:
                                                ;
                                                ;CHECK FOR ERRORS DURING WRITE OPERATION
030332  004737  041126                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
030336  000405                                  BR      13$             ;GO TO 13$ IF NO ERROR
030340  000240                                  NOP                     ;RETURN HERE IF ERROR
030342  104000                                  EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
030344  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030346  000137  030664                          JMP     25$             ;GO TO 25$ IF ERROR
030352                                  13$:
030352  004737  054166                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
030356  000405                                  BR      14$             ;GO TO 14$ IF NO ERROR
030360  000240                                  NOP                     ;RETURN HERE IF ERROR
030362  104000                                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
030364  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030366  000137  030664                          JMP     25$             ;GO TO 25$ IF ERROR
030372                                  14$:
030372  004737  041760                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
030376  000405                                  BR      15$             ;GO TO 15$ IF NO ERROR
030400  000240                                  NOP                     ;RETURN HERE IF ERROR
030402  104000                                  EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
030404  004736                                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030406  000137  030664                          JMP     25$             ;GO TO 25$ IF ERROR
030412                                  15$:
                                                ;
                                                ;CHANGE LOOP ADDRESSES
030412  012737  030422  001124                  MOV     #16$,$LPERR
030420  000410                                  BR      17$
                                                ;
                                                ;;*****************************************************************
                                                ;LOOP #3        READ
                                                ;
030422                                  16$:
                                                ;
                                                ;PREPARE DEVICE FOR READ OPERATION
030422  004737  034032                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
030426  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
030430  000404                    BR      17$           ;GO TO 17$ IF NO ERROR
030432  000240                    NOP                   ;RETURN HERE IF ERROR
030434  104000                    EMT                   ;ERROR # DEFINED BY TSTPRP SUBROUTINE
030436  000137  030664            JMP     25$           ;GO TO 25$ IF ERROR
2454 030442                     17$:
2455
2456                            ;READ DATA FROM DEVICE
2457 030442                     18$:
2458 030442  012737  000071  001412    MOV     #RD:GO,RMCS10         ;READ DATA COMMAND
2459 030450  012737  105264  001416    MOV     #BUFTWO+4,RMBA0       ;CHANGE BUFFER
2460 030456  012702  001555            MOV     #PUTINX,R2            ;LOAD PUT REGISTER INDEX TABLE
2461 030462  112722  000006            MOVB    #RMDA,(R2)+
2462 030466  112722  000032            MOVB    #RMOF,(R2)+
2463 030472  112722  000034            MOVB    #RMDC,(R2)+
2464 030476  112722  000004            MOVB    #RMBA,(R2)+
2465 030502  112722  000002            MOVB    #RMWC,(R2)+
2466 030506  112722  000000            MOVB    #RMCS1,(R2)+
2467 030512  112712  000200            MOVB    #200,(R2)
2468
2469 030516  004737  040360            JSR     PC,PUT        ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
030522  000404                    BR      19$           ;GO TO 19$ IF NO ERROR
030524  000240                    NOP                   ;RETURN HERE IF ERROR
030526  104000                    EMT                   ;ERROR # DEFINED BY PUT SUBROUTINE
030530  000137  030664            JMP     25$           ;GO TO 25$ IF ERROR
2470 030534                     19$:
2471
2472                            ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
030534  004737  040024            JSR     PC,GETSTS     ;GO TO GETSTS SUBROUTINE
2473
2474                            ;WAIT FOR COMMAND TO COMPLETE
030540  004737  040732            JSR     PC,TIMOUT     ;GO TO TIMOUT SUBROUTINE
2475
2476                            ;GO READ STATUS FOR READ OPERATION
2477 030544  004737  040110            JSR     PC,GET        ;GO READ REGISTER(S) WITH GET SUBROUTINE
030550  000404                    BR      20$           ;GO TO 20$ IF NO ERROR
030552  000240                    NOP                   ;RETURN HERE IF ERROR
030554  104000                    EMT                   ;ERROR # DEFINED BY GET SUBROUTINE
030556  000137  030664            JMP     25$           ;GO TO 25$ IF ERROR
2478 030562                     20$:
2479
2480                            ;CHECK FOR ERRORS DURING READ OPERATION
2481 030562  004737  041126            JSR     PC,PRIERR     ;GO CHECK FOR PRIMARY ERRORS
030566  000405                    BR      21$           ;GO TO 21$ IF NO ERROR
030570  000240                    NOP                   ;RETURN HERE IF ERROR
030572  104000                    EMT                   ;ERROR # DEFINED BY PRIERR SUBROUTINE
030574  004736                    JSR     PC,@(SP)+     ;GO BACK FOR MORE ERROR CHECKS
030576  000137  030664            JMP     25$           ;GO TO 25$ IF ERROR
2482 030602                     21$:
2483 030602  004737  054166            JSR     PC,DTASTS     ;GO VERIFY RESULTS OF DATA TRANSFER
030606  000405                    BR      22$           ;GO TO 22$ IF NO ERROR
```

```
030610  000240                          NOP                     ;RETURN HERE IF ERROR
030612  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
030614  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030616  000137  030664                  JMP     25$             ;GO TO 25$ IF ERROR
2484 030622                     22$:
2485 030622  004737  041760            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
030626  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
030630  000240                          NOP                     ;RETURN HERE IF ERROR
030632  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
030634  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
030636  000137  030664                  JMP     25$             ;GO TO 25$ IF ERROR
2486 030642                     23$:
2487 030642  004737  037462            JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
030646  104260                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
030650  105264                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
030652  000404                          BR      24$             ;GO TO 24$ IF NO ERROR
030654  000240                          NOP                     ;RETURN HERE IF ERROR
030656  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
030660  000137  030664                  JMP     25$             ;GO TO 25$ IF ERROR
2488 030664                     24$:
2489
2490 030664                     25$:
2491
2492                            ;;*******************************************************************
                                ;*TEST 21        WRITE, READ EACH TRACK W/ MIXED PEAK SHIFT
                                ;;*******************************************************************
030664                          TST21:
030664  000004                          SCOPE                   ;SCOPE CALL
030666  000240                          NOP                     ;START OF TEST
030670  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
030674  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
030700  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
030704  012737  000021  001226          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2493
2494                            ;;*******************************************************************
2495                            ;LOOP #1        FORMAT,WRITE,READ
2496
2497 030712                     1$:
2498
2499                            ;PREPARE THE DEVICE FOR FORMAT OPERATION
2500 030712  004737  034032            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
030716  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
030720  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
030722  000240                          NOP                     ;RETURN HERE IF ERROR
030724  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
030726  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
2501 030732                     2$:
2502
2503                            ;LOAD PARAMETERS AND GENERATE DATA BUFFER
2504 030732  012737  001060  001446          MOV     #560.,RMDCO             ;CYLINDER = 560.
```

```
2505 030740  012737  000000  001420          MOV     #0,RMDAO                ;TRACK = 0, SECTOR = 0
2506 030746                          3$:
2507 030746  012737  104254  001416          MOV     #BUFONE,RMBAO           ;BUS ADDRESS
2508 030754  012737  177376  001414          MOV     #-258.,RMWCO            ;2 + 256. WORDS (2'S COMP)
2509 030762  012737  010000  001444          MOV     #FMT16,RMOFO            ;16 BIT FORMAT
2510 030770  012737  000063  001412          MOV     #WH!GO,RMCS1O           ;WRITE HEADER AND DATA COMMAND
2511
2512                                  ;VERIFY THAT SECTOR IS NOT BAD
     030776  004737  034762                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     031002  000405                          BR      4$              ;GO TO 4$ IF NO ERROR
     031004  104401  066006                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     031010  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     031012  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
2513 031016                          4$:
2514 031016  012737  067354  001174          MOV     #MIXED,$TMP0            ;STARTING ADDRESS OF PATTERN
2515 031024  012737  000200  001176          MOV     #128.,$TMP1             ;RANGE OF PATTERN
2516 031032  004737  037214                  JSR     PC,GENBUF               ;GO GENERATE BUFFER FOR FORMAT
2517
2518                                  ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATIION
2519 031036  012702  001555          MOV     #PUTINX,R2              ;R2 = BYTE ENTRY POSITION
2520 031042  112722  000034          MOVB    #RMDC,(R2)+
2521 031046  112722  000006          MOVB    #RMDA,(R2)+
2522 031052  112722  000004          MOVB    #RMBA,(R2)+
2523 031056  112722  000002          MOVB    #RMWC,(R2)+
2524 031062  112722  000032          MOVB    #RMOF,(R2)+
2525 031066  112722  000000          MOVB    #RMCS1,(R2)+
2526 031072  112712  000200          MOVB    #200,(R2)               ;TERMINATE TABLE
2527 031076                          5$:
2528
2529                                  ;FORMAT THE DRIVE
2530 031076  004737  040360          JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031102  000404                  BR      6$              ;GO TO 6$ IF NO ERROR
     031104  000240                  NOP                     ;RETURN HERE IF ERROR
     031106  104000                  EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     031110  000137  031712          JMP     26$             ;GO TO 26$ IF ERROR
2531 031114                          6$:
2532
2533                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     031114  004737  040024          JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2534
2535                                  ;WAIT FOR COMMAND TO COMPLETE
     031120  004737  040732          JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2536
2537                                  ;GO READ STATUS FOR FORMAT OPERATION
2538 031124  004737  040110          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     031130  000404                  BR      7$              ;GO TO 7$ IF NO ERROR
     031132  000240                  NOP                     ;RETURN HERE IF ERROR
     031134  104000                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     031136  000137  031712          JMP     26$             ;GO TO 26$ IF ERROR
2539 031142                          7$:
2540
2541                                  ;VERIFY NO ERRORS DURING FORMAT
2542 031142  004737  054166          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     031146  000405                  BR      8$              ;GO TO 8$ IF NO ERROR
     031150  000240                  NOP                     ;RETURN HERE IF ERROR
     031152  104000                  EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     031154  004736                  JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
      031156  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2543 031162                           8$:
 2544
 2545                                  ;MOVE LOOP ADDRESSES TO NEXT OPERATION
 2546 031162  012737  031172  001124          MOV     #9$,$LPERR
 2547 031170  000410                          BR      10$                     ;SKIP TO WRITE OPERATION
 2548
 2549                                  ;;*****************************************************************
 2550                                  ;LOOP #2        WRITE,READ
 2551
 2552 031172                           9$:
 2553
 2554                                  ;PREPARE DEVICE FOR WRITE OPERATION
 2555 031172  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      031176  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                      ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                      ;CLEAR CONTROLLER & SELECT DEVICE
                                                                      ;VERIFY CONTROLLER CLEAR OPERATION
                                                                      ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                      ;VERIFY PACK ACKNOWLEDGE
                                                                      ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                      ;VERIFY RECALIBRATION
      031200  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
      031202  000240                          NOP                     ;RETURN HERE IF ERROR
      031204  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      031206  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2556 031212                           10$:
 2557
 2558                                  ;WRITE DATA TO THE DRIVE
 2559 031212                           11$:
 2560 031212  012737  104260  001416          MOV     #BUFONE+4,RMBAO         ;CHANGF BUS ADDRESS
 2561 031220  012737  177400  001414          MOV     #-256.,RMWCO    ;CHANGE WORD COUNT
 2562 031226  012737  000061  001412          MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND
 2563 031234  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2564 031240  112722  000006                  MOVB    #RMDA,(R2)+
 2565 031244  112722  000034                  MOVB    #RMDC,(R2)+
 2566 031250  112722  000032                  MOVB    #RMOF,(R2)+
 2567 031254  112722  000004                  MOVB    #RMBA,(R2)+
 2568 031260  112722  000002                  MOVB    #RMWC,(R2)+
 2569 031264  112722  000000                  MOVB    #RMCS1,(R2)+
 2570 031270  112722  000200                  MOVB    #200,(R2)+              ;TERMINATE TABLE
 2571
 2572 031274  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      031300  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
      031302  000240                          NOP                     ;RETURN HERE IF ERROR
      031304  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      031306  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2573 031312                           12$:
 2574
 2575                                  ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      031312  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2576
 2577                                  ;WAIT FOR COMMAND TO COMPLETE
      031316  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2578
 2579                                  ;GO READ STATUS FOR WRITE COMMAND
 2580 031322  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
```

T21 WRITE, READ EACH TRACK W/ MIXED PEAK SHIFT

```
        031326  000404                          BR      13$             ;GO TO 13$ IF NO ERROR
        031330  000240                          NOP                     ;RETURN HERE IF ERROR
        031332  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        031334  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2581   031340                          13$:
 2582
 2583                                   ;CHECK FOR ERRORS DURING WRITE OPERATION
 2584   031340  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        031344  000405                          BR      14$             ;GO TO 14$ IF NO ERROR
        031346  000240                          NOP                     ;RETURN HERE IF ERROR
        031350  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        031352  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031354  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2585   031360                          14$:
 2586   031360  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
        031364  000405                          BR      15$             ;GO TO 15$ IF NO ERROR
        031366  000240                          NOP                     ;RETURN HERE IF ERROR
        031370  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
        031372  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031374  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2587   031400                          15$:
 2588   031400  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        031404  000405                          BR      16$             ;GO TO 16$ IF NO ERROR
        031406  000240                          NOP                     ;RETURN HERE IF ERROR
        031410  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        031412  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        031414  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2589   031420                          16$:
 2590
 2591                                   ;CHANGE LOOP ADDRESSES
 2592   031420  012737  031430  001124          MOV     #17$,$LPERR
 2593   031426  000410                          BR      18$                             ;SKIP TO NEXT OPERATION
 2594
 2595                                   ;;*****************************************************************
 2596                                   ;LOOP #3        READ
 2597
 2598   031430                          17$:
 2599
 2600                                   ;PREPARE DEVICE FOR READ OPERATION
 2601   031430  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        031434  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        031436  000404                          BR      18$             ;GO TO 18$ IF NO ERROR
        031440  000240                          NOP                     ;RETURN HERE IF ERROR
        031442  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        031444  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2602   031450                          18$:
 2603
 2604                                   ;READ DATA FROM DEVICE
 2605   031450                          19$:
 2606   031450  012737  105264  001416          MOV     #BUFTWO+4,RMBAO         ;CHANGE BUS ADDRESS
```

```
 2607 031456  012737  000071  001412          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
 2608 031464  012702  001555                  MOV     #PUTINX,R2              ;LOAD PUT REGISTER INDEX TABLE
 2609 031470  112722  000006                  MOVB    #RMDA,(R2)+
 2610 031474  112722  000032                  MOVB    #RMOF,(R2)+
 2611 031500  112722  000034                  MOVB    #RMDC,(R2)+
 2612 031504  112722  000004                  MOVB    #RMBA,(R2)+
 2613 031510  112722  000002                  MOVB    #RMWC,(R2)+
 2614 031514  112722  000000                  MOVB    #RMCS1,(R2)+
 2615 031520  112712  000200                  MOVB    #200,(R2)
 2616
 2617 031524  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      031530  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
      031532  000240                          NOP                     ;RETURN HERE IF ERROR
      031534  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      031536  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2618 031542                          20$:
 2619
 2620                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
      031542  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
 2621
 2622                                 ;WAIT FOR COMMAND TO COMPLETE
      031546  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
 2623
 2624                                 ;GO READ STATUS FOR READ OPERATION
 2625 031552  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      031556  000404                          BR      21$             ;GO TO 21$ IF NO ERROR
      031560  000240                          NOP                     ;RETURN HERE IF ERROR
      031562  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      031564  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2626 031570                          21$:
 2627
 2628                                 ;CHECK FOR ERRORS DURING READ OPERATION
 2629 031570  004737  041126                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      031574  000405                          BR      22$             ;GO TO 22$ IF NO ERROR
      031576  000240                          NOP                     ;RETURN HERE IF ERROR
      031600  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      031602  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      031604  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2630 031610                          22$:
 2631 031610  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
      031614  000405                          BR      23$             ;GO TO 23$ IF NO ERROR
      031616  000240                          NOP                     ;RETURN HERE IF ERROR
      031620  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
      031622  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      031624  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2632 031630                          23$:
 2633 031630  004737  041760                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      031634  000405                          BR      24$             ;GO TO 24$ IF NO ERROR
      031636  000240                          NOP                     ;RETURN HERE IF ERROR
      031640  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      031642  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      031644  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
 2634 031650                          24$:
 2635 031650  004737  037462                  JSR     PC,CMPBUF       ;GO COMPARE WRITE, READ DATA BUFFERS
      031654  104260                          .WORD   BUFONE+4                ;STARTING ADDRESS OF WRITE BUFFER
      031656  105264                          .WORD   BUFTWO+4                ;STARTING ADDRESS OF READ BUFFER
      031660  000404                          BR      25$             ;GO TO 25$ IF NO ERROR
```

```
        031662  000240                          NOP                     ;RETURN HERE IF ERROR
        031664  104000                          EMT                     ;ERROR # DEFINED BY CMPBUF SUBROUTINE
        031666  000137  031712                  JMP     26$             ;GO TO 26$ IF ERROR
  2636  031672                          25$:
  2637  031672  105237  001421                  INCB    RMDAO+1                 ;ADVANCE TO NEXT TRACK
  2638  031676  123737  001421  001335          CMPB    RMDAO+1,LSTRK+1         ;DONE ?
  2639  031704  101002                          BHI     26$                     ;YES!!
  2640  031706  000137  030746                  JMP     3$                      ;TEST NEXT TRACK
  2641  031712                          26$:
  2642
  2643                                  ;;*******************************************************************
                                        ;*TEST 22        READ, WRITE CHECK MULTIPLE SECTORS IN OFFSET MODE
                                        ;;*******************************************************************
        031712                          TST22:
        031712  000004                          SCOPE                   ;SCOPE CALL
        031714  000240                          NOP                     ;START OF TEST
        031716  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        031722  013700  001276                  MOV     @BASE,R0        ;R0 = UNIBUS ADDRESS
        031726  013701  001466                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        031732  012737  000022  001226          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  2644
  2645                                  ;;*******************************************************************
                                        ;LOOP #1        FORMAT,READ OFFSET,WRITE CHECK OFFSET IN BOTH DIRECTIONS
  2647  031740                          1$:
  2648
  2649                                  ;PREPARE DEVICE FOR FORMAT OPERATION
  2650  031740  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        031744  154130                          .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        031746  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
        031750  000240                          NOP                     ;RETURN HERE IF ERROR
        031752  104000                          EMT                     ;ERROR # DEFINED BY TSTPKP SUBROUTINE
        031754  000137  033074                  JMP     28$             ;GO TO 28$ IF ERROR
  2651
  2652                                  ;LOAD PARAMETERS AND GENERATE DATA BUFFER
  2653  031760                          2$:
  2654  031760  012737  001057  001446          MOV     #559.,RMDCO             ;CYLINDER = 559.
  2655  031766  012737  001000  001420          MOV     #TA2,RMDAO              ;TRACK = 2, SECTOR = 0
  2656  031774  012737  104254  001416          MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS
  2657  032002  012737  176774  001414          MOV     #-258.*2,RMWCO          ;WORD COUNT FOR 2 SECTORS (2'S COMP)
  2658  032010  012737  010000  001444          MOV     #FMT16,RMOFO            ;16 BIT MODE
  2659  032016  012737  000063  001412          MOV     #WH!GO,RMCS10           ;WRITE HEADER
  2660
  2661                                  ;VERIFY THAT SECTOR IS NOT BAD
        032024  004737  034762                  JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
        032030  000405                          BR      3$              ;GO TO 3$ IF NO ERROR
        032032  104401  066006                  TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
        032036  104000                          EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
        032040  000137  033074                  JMP     28$             ;GO TO 28$ IF ERROR
  2662  032044                          3$:
  2663  032044  012737  067456  001174          MOV     #ZEROS,$TMP0            ;DATA PATTERN
```

```
2664 032052  012737  000001  001176          MOV     #1,$TMP1
2665 032060  004737  037214                  JSR     PC,GENBUF               ;TO GENERATE BUFFER FOR FORMAT
2666
2667                                 ;LOAD PUT REGISTER INDEX TABLE FOR FORMAT OPERATION
2668 032064  012702  001555                  MOV     #PUTINX,R2              ;BYTE ENTRY TABLE
2669 032070  112722  000034                  MOVB    #RMDC,(R2)+
2670 032074  112722  000006                  MOVB    #RMDA,(R2)+
2671 032100  112722  000004                  MOVB    #RMBA,(R2)+
2672 032104  112722  000002                  MOVB    #RMWC,(R2)+
2673 032110  112722  000032                  MOVB    #RMOF,(R2)+
2674 032114  112722  000000                  MOVB    #RMCS1,(R2)+
2675 032120  112712  000200                  MOVB    #200,(R2)               ;TABLE TERMINATOR
2676 032124                          4$:
2677 032124  004737  040360                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032130  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
     032132  000240                          NOP                     ;RETURN HERE IF ERROR
     032134  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032136  000137  033074                  JMP     28$             ;GO TO 28$ IF ERROR
2678 032142                          5$:
2679
2680                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     032142  004737  040024                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2681
2682                                 ;WAIT FOR COMMAND TO COMPLETE
     032146  004737  040732                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2683
2684                                 ;GO READ STATUS FOR FORMAT OPERATION
2685 032152  004737  040110                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032156  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
     032160  000240                          NOP                     ;RETURN HERE IF ERROR
     032162  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032164  000137  033074                  JMP     28$             ;GO TO 28$ IF ERROR
2686 032170                          6$:
2687
2688                                 ;VERIFY NO ERROR DURING FORMAT
2689 032170  004737  054166                  JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     032174  000405                          BR      7$              ;GO TO 7$ IF NO ERROR
     032176  000240                          NOP                     ;RETURN HERE IF ERROR
     032200  104000                          EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     032202  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032204  000137  033074                  JMP     28$             ;GO TO 28$ IF ERROR
2690
2691                                 ;MOV LOOP ADDRESSES TO NEXT OPERATION
2692 032210                          7$:
2693 032210  012737  032232  001124          MOV     #8$,$LPERR      ;ERROR LOOP ADDRESS
2694
2695                                 ;LOCATE BUFFER ADDRESS AND WORD COUNT
2696 032216  012737  177000  001414          MOV     #-256.*2,RMWCO  ;TWO SECTORS
2697 032224  012737  104254  001416          MOV     #BUFONE,RMBAO           ;BUFFER ADDRESS
2698                                 ;;*****************************************************************
2699                                 ;LOOP 2  READ AND WRITE CHECK IN OFFSET MODE
2700 032232                          8$:
2701                                 ;PREPARE DEVICE FOR READ OFFSET OPEATION
2702 032232  004737  034032                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     032236  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
     032240  000404                        BR      9$          ;GO TO 9$ IF NO ERROR
     032242  000240                        NOP                 ;RETURN HERE IF ERROR
     032244  104000                        EMT                 ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     032246  000137  033074                JMP     28$         ;GO TO 28$ IF ERROR
2703 032252                         9$:
2704 032252                         10$:
2705                                ;READ OFFSET
2706 032252  012702  001555                MOV     #PUTINX,R2  ;LOAD THE TABLE
2707 032256  112722  000006                MOVB    #RMDA,(R2)+
2708 032262  112722  000034                MOVB    #RMDC,(R2)+
2709 032266  112722  000032                MOVB    #RMOF,(R2)+
2710 032272  112722  000004                MOVB    #RMBA,(R2)+
2711 032276  112722  000002                MOVB    #RMWC,(R2)+
2712 032302  112722  000200                MOVB    #200,(R2)+  ;TABLE TERMINATOR
2713
2714 032306  004737  040360                JSR     PC,PUT      ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032312  000404                        BR      11$         ;GO TO 11$ IF NO ERROR
     032314  000240                        NOP                 ;RETURN HERE IF ERROR
     032316  104000                        EMT                 ;ERROR # DEFINED BY PUT SUBROUTINE
     032320  000137  033074                JMP     28$         ;GO TO 28$ IF ERROR
2715 032324  012737  000015  001412  11$:  MOV     #OFFSET!GO,RMCS10       ;OFFSET COMMAND
2716 032332  012702  001555                MOV     #PUTINX,R2
2717 032336  112722  000000                MOVB    #RMCS1,(R2)+
2718 032342  112722  000200                MOVB    #200,(R2)+
2719
2720 032346  004737  040360                JSR     PC,PUT      ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032352  000404                        BR      12$         ;GO TO 12$ IF NO ERROR
     032354  000240                        NOP                 ;RETURN HERE IF ERROR
     032356  104000                        EMT                 ;ERROR # DEFINED BY PUT SUBROUTINE
     032360  000137  033074                JMP     28$         ;GO TO 28$ IF ERROR
2721 032364                         12$:
2722
2723                                ;WAIT FOR COMMAND TO COMPLETE
     032364  004737  040732                JSR     PC,TIMOUT   ;GO TO TIMOUT SUBROUTINE
2724
2725 032370  012737  000071  001412        MOV     #RD!GO,RMCS10       ;READ DATA COMMAND
2726 032376  012702  001555                MOV     #PUTINX,R2          ;TABLE INDEX
2727 032402  112722  000000                MOVB    #RMCS1,(R2)+
2728 032406  112712  000200                MOVB    #200,(R2)           ;TABLE TERMINATOR
2729
2730 032412  004737  040360                JSR     PC,PUT      ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032416  000404                        BR      13$         ;GO TO 13$ IF NO ERROR
     032420  000240                        NOP                 ;RETURN HERE IF ERROR
     032422  104000                        EMT                 ;ERROR # DEFINED BY PUT SUBROUTINE
     032424  000137  033074                JMP     28$         ;GO TO 28$ IF ERROR
2731 032430                         13$:
2732
2733                                ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     032430  004737  040024                JSR     PC,GETSTS   ;GO TO GETSTS SUBROUTINE
2734
2735                                ;WAIT FOR COMMAND TO COMPLETE
     032434  004737  040732                JSR     PC,TIMOUT   ;GO TO TIMOUT SUBROUTINE
```

```
2736
2737                                    ;READ STATUS FOR READ OFFSET OPERATION
2738 032440  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032444  000404                                BR      14$             ;GO TO 14$ IF NO ERROR
     032446  000240                                NOP                     ;RETURN HERE IF ERROR
     032450  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032452  000137  033074                        JMP     28$             ;GO TO 28$ IF ERROR
2739
2740                                    ;CHECK ERROR DURING READ OFFSET
2741 032456                             14$:
2742 032456  004737  041126                        JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     032462  000405                                BR      15$             ;GO TO 15$ IF NO ERROR
     032464  000240                                NOP                     ;RETURN HERE IF ERROR
     032466  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     032470  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032472  000137  033074                        JMP     28$             ;GO TO 28$ IF ERROR
2743 032476                             15$:
2744 032476  004737  054166                        JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     032502  000405                                BR      16$             ;GO TO 16$ IF NO ERROR
     032504  000240                                NOP                     ;RETURN HERE IF ERROR
     032506  104000                                EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     032510  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032512  000137  033074                        JMP     28$             ;GO TO 28$ IF ERROR
2745 032516                             16$:
2746 032516  004737  041760                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     032522  000405                                BR      17$             ;GO TO 17$ IF NO ERROR
     032524  000240                                NOP                     ;RETURN HERE IF ERROR
     032526  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     032530  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032532  000137  033074                        JMP     28$             ;GO TO 28$ IF ERROR
2747 032536                             17$:
2748
2749                                    ;CHANGE LOOP ADDRESS
2750 032536  012737  032546  001124                MOV     #18$,$LPERR
2751 032544  000410                                BR      19$             ;TO NEXT OPERATION
2752
2753                                    ;;*****************************************************************
2754                                    ;LOOP 3 WRITE CHECK OFFSET
2755 032546                             18$:
2756
2757                                    ;PREPARE DEVICE FOR READ OPERATION
2758 032546  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     032552  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                            ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                            ;VERIFY PACK ACKNOWLEDGE
                                                                            ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                            ;VERIFY RECALIBRATION
     032554  000404                                BR      19$             ;GO TO 19$ IF NO ERROR
     032556  000240                                NOP                     ;RETURN HERE IF ERROR
     032560  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     032562  000137  033074                        JMP     28$             ;GO TO 28$ IF ERROR
2759 032566                             19$:
2760
2761                                    ;WRITE CHECK DATA IN OFFSET MODE
```

```
2762 032566                                    20$:
2763 032566  012702  001555                            MOV     #PUTINX,R2
2764 032572  112722  000006                            MOVB    #RMDA,(R2)+
2765 032576  112722  000032                            MOVB    #RMOF,(R2)+
2766 032602  112722  000034                            MOVB    #RMDC,(R2)+
2767 032606  112722  000004                            MOVB    #RMBA,(R2)+
2768 032612  112722  0000C2                            MOVB    #RMWC,(R2)+
2769 032616  112712  000200                            MOVB    #200,(R2)       ;TABLE TERMINATOR
2770
2771 032622  004737  040360                            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032626  000404                                    BR      21$             ;GO TO 21$ IF NO ERROR
     032630  000240                                    NOP                     ;RETURN HERE IF ERROR
     032632  104000                                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032634  000137  033074                            JMP     28$             ;GO TO 28$ IF ERROR
2772
2773                                           ;SETUP OFFSET MODE FIRST
2774 032640                                    21$:
2775 032640  012737  000015  001412                    MOV     #OFFSET!GO,RMCS10       ;OUTPUT BUFFER
2776 032646  012702  001555                            MOV     #PUTINX,R2
2777 032652  112722  000000                            MOVB    #RMCS1,(R2)+
2778 032656  112712  000200                            MOVB    #200,(R2)
2779
2780 032662  004737  040360                            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032666  000404                                    BR      22$             ;GO TO 22$ IF NO ERROR
     032670  000240                                    NOP                     ;RETURN HERE IF ERROR
     032672  104000                                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032674  000137  033074                            JMP     28$             ;GO TO 28$ IF ERROR
2781 032700                                    22$:
2782
2783                                           ;WAIT FOR COMMAND TO COMPLETE
     032700  004737  040732                            JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2784
2785                                           ;EXECUTE WRITE CHECK DATA COMMAND
2786 032704  012737  000051  001412                    MOV     #WCD!GO,RMCS10
2787 032712  012702  001555                            MOV     #PUTINX,R2
2788 032716  112722  000000                            MOVB    #RMCS1,(R2)+
2789 032722  112712  000200                            MOVB    #200,(R2)
2790
2791 032726  004737  040360                            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032732  000404                                    BR      23$             ;GO TO 23$ IF NO ERROR
     032734  000240                                    NOP                     ;RETURN HERE IF ERROR
     032736  104000                                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032740  000137  033074                            JMP     28$             ;GO TO 28$ IF ERROR
2792 032744                                    23$:
2793
2794                                           ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     032744  004737  040024                            JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2795
2796                                           ;WAIT FOR COMMAND TO COMPLETE
     032750  004737  040732                            JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2797
2798                                           ;READ STATUS FOR WRITE CHECK OFFSET OPERATION
2799 032754  004737  040110                            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032760  000404                                    BR      24$             ;GO TO 24$ IF NO ERROR
     032762  000240                                    NOP                     ;RETURN HERE IF ERROR
     032764  104000                                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032766  000137  033074                            JMP     28$             ;GO TO 28$ IF ERROR
```

```
2800 032772                                    24$:
2801
2802                                           ;CHECK ERROR DURING WRITE CHECK OFFSET OPERATION
2803 032772  004737  041126                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     032776  000405                                 BR      25$             ;GO TO 25$ IF NO ERROR
     033000  000240                                 NOP                     ;RETURN HERE IF ERROR
     033002  104000                                 EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     033004  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033006  000137  033074                         JMP     28$             ;GO TO 28$ IF ERROR
2804 033012                                    25$:
2805 033012  004737  054166                          JSR     PC,DTASTS       ;GO VERIFY RESULTS OF DATA TRANSFER
     033016  000405                                 BR      26$             ;GO TO 26$ IF NO ERROR
     033020  000240                                 NOP                     ;RETURN HERE IF ERROR
     033022  104000                                 EMT                     ;ERROR # DEFINED BY DTASTS SUBROUTINE
     033024  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033026  000137  033074                         JMP     28$             ;GO TO 28$ IF ERROR
2806 033032                                    26$:
2807 033032  004737  041760                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     033036  000405                                 BR      27$             ;GO TO 27$ IF NO ERROR
     033040  000240                                 NOP                     ;RETURN HERE IF ERROR
     033042  104000                                 EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     033044  004736                                 JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033046  000137  033074                         JMP     28$             ;GO TO 28$ IF ERROR
2808 033052                                    27$:
2809 033052  032737  000200  001444                BIT     #OFD,RMOFO      ;IN FORWARD DIRECTION OF OFFSET ?
2810 033060  001005                                 BNE     28$             ;BRANCH IF SO
2811 033062  052737  000200  001444                BIS     #OFD,RMOFO      ;SET TO FORWARD DIRECTION
2812 033070  000137  032210                         JMP     7$              ;CHANGE OFFSET DIRECTION TEST AGAIN
2813 033074                                    28$:
2814
2815                                           ;;*****************************************************************
                                               ;*TEST 23       FORMAT FE CYLINDERS
                                               ;;*****************************************************************
     033074                                    TST23:
     033074  000004                                 SCOPE                   ;SCOPE CALL
     033076  000240                                 NOP                     ;START OF TEST
     033100  012706  001100                         MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     033104  013700  001276                         MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     033110  013701  001466                         MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     033114  012737  000001  001206                 MOV     #1,$TIMES       ;;DO 1 ITERATION
     033122  012737  000023  001226                 MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2816
2817                                           ;SETUP PARAMETERS FOR GENERATING DATA BUFFER
2818 033130  012737  001057  001446                MOV     #559.,RMDC0     ;CYLINDER = 559.
2819 033136  012737  001000  001420                MOV     #TA2,RMDA0      ;TRACK = 2, SECTOR = 0
2820 033144                                    1$:
2821 033144  012737  011000  001444                MOV     #FMT16!SSEI,RMOFO       ;16 BIT FORMAT WITH SSEI SET
2822 033152  012737  177776  001414                MOV     #-2,RMWC0       ;2 WORDS (2'S COMP)
2823 03316C  012737  104254  001416                MOV     #BUFONE,RMBA0   ;DATA BUFFER ADDRESS
2824 033166  012737  000062  001412                MOV     #WH,RMCS10      ;WRITE HEADER AND DATA
2825
2826                                           ;VERIFY THAT SECTOR IS NOT BAD
     033174  004737  034762                         JSR     PC,BADSCT       ;CALL BAD SECTOR MODULE
     033200  000405                                 BR      2$              ;GO TO 2$ IF NO ERROR
     033202  104401  066006                         TYPE    ,SCTMSG         ;TYPE BAD SECTOR MESSAGE
     033206  104000                                 EMT                     ;ERROR # DEFINED BY BADSCT SUBROUTINE
     033210  000137  033522                         JMP     12$             ;GO TO 12$ IF ERROR
```

```
2827 033214                                    2$:
2828 033214  023727  001446  001060                CMP     RMDCO,#560.     ;IS DESIRED CYLINDER, OUTSIDE FE CYLINDERS ?
2829 033222  101402                                BLOS    3$              ;NO !!
2830 033224  000137  033522                        JMP     12$             ;YES, EXIT TEST
2831 033230                                    3$:
2832 033230  004737  037214                        JSR     PC,GENBUF       ;GO GENERATE DATA BUFFER
2833 033234                                    4$:
2834 033234  005737  001514                        TST     SSFENB          ;TEST SSF ENABLE FOR 1ST HEADER WORD
2835 033240  001403                                BEQ     5$              ;NO, DO NOT SET "SSF"
2836 033242  052737  020000  104254                BIS     #SSF,BUFONE     ;YES, SET SSF BIT
2837 033250                                    5$:
2838
2839                                           ;PREPARE DEVICE FOR DATA TRANSFER
2840 033250  004737  034032                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     033254  154130                                .WORD   154130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                           ;SELECT DEVICE & VERIFY DEVICE AVAILABLE
                                                                           ;CLEAR CONTROLLER & SELECT DEVICE
                                                                           ;VERIFY CONTROLLER CLEAR OPERATION
                                                                           ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                           ;VERIFY PACK ACKNOWLEDGE
                                                                           ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                           ;VERIFY RECALIBRATION
     033256  000404                                BR      6$              ;GO TO 6$ IF NO ERROR
     033260  000240                                NOP                     ;RETURN HERE IF ERROR
     033262  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     033264  000137  033522                        JMP     12$             ;GO TO 12$ IF ERROR
2841
2842                                           ;SETUP AND EXECUTE WRITE HEADER AND DATA COMMAND
2843 033270                                    6$:
2844 033270  012737  000063  001412                MOV     #WH!GO,RMCS10   ;WRITE HEADER AND DATA
2845 033276  012702  001555                        MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
2846 033302  112722  000006                        MOVB    #RMDA,(R2)+
2847 033306  112722  000034                        MOVB    #RMDC,(R2)+
2848 033312  112722  000032                        MOVB    #RMOF,(R2)+
2349 033316  112722  000002                        MOVB    #RMWC,(R2)+
2850 033322  112722  000004                        MOVB    #RMBA,(R2)+
2851 033326  112722  000030                        MOVB    #RMCS1,(R2)+
2852 033332  112722  000200                        MOVB    #200,(R2)+
2853
2854 033336  004737  040360                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     033342  000404                                BR      7$              ;GO TO 7$ IF NO ERROR
     033344  000240                                NOP                     ;RETURN HERE IF ERROR
     033346  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     033350  000137  033522                        JMP     12$             ;GO TO 12$ IF ERROR
2855 033354                                    7$:
2856
2857                                           ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     033354  004737  040024                        JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
2858
2859                                           ;WAIT FOR COMMAND TO COMPLETE
     033360  004737  040732                        JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
2860
2861 033364  004737  040110                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     033370  000404                                BR      8$              ;GO TO 8$ IF NO ERROR
     033372  000240                                NOP                     ;RETURN HERE IF ERROR
     033374  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     033376  000137  033522                        JMP     12$             ;GO TO 12$ IF ERROR
```

```
2862 033402                                    8$:
2863
2864                                           ;VERIFY RESULTS OF WRITE COMMAND
2865 033402  004737  041126                          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     033406  000405                                BR      9$              ;GO TO 9$ IF NO ERROR
     033410  000240                                NOP                     ;RETURN HERE IF ERROR
     033412  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     033414  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033416  000137  033522                        JMP     12$             ;GO TO 12$ IF ERROR
2866 033422                                    9$:
2867 033422  004737  041760                          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     033426  000405                                BR      10$             ;GO TO 10$ IF NO ERROR
     033430  000240                                NOP                     ;RETURN HERE IF ERROR
     033432  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     033434  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033436  000137  033522                        JMP     12$             ;GO TO 12$ IF ERROR
2868 033442                                    10$:
2869
2870                                           ;INCREMENT ADDRESS
2871 033442  005237  001420                          INC     RMDA0           ;ADVANCE SECTOR COUNT
2872 033446  123727  001420  000037                  CMPB    RMDA0,#31.      ;DONE ALL SECTORS ?
2873 033454  101420                                BLOS    11$             ;NO !!
2874
2875 033456  105037  001420                          CLRB    RMDA0           ;START AT SECTOR 0 AND
2876 033462  105237  001421                          INCB    RMDA0+1         ;ADVANCE TO NEXT TRACK.
2877 033466  123737  001421  001335                  CMPB    RMDA0+1,LSTRK+1 ;DONE WITH LAST TRACK ?
2878 033474  101410                                BLOS    11$             ;NO !!
2879
2880 033476  105037  001421                          CLRB    RMDA0+1         ;START WITH TRACK 0 AND
2881 033502  005237  001446                          INC     RMDC0           ;ADVANCE TO NEXT CYLINDER.
2882 033506  023727  001446  001060                  CMP     RMDC0,#560.     ;DONE WITH LAST CYLINDER ?
2883 033514  101002                                BHI     12$             ;YES !!
2884 033516  000137  033144            11$:        JMP     1$              ;GO DO TEST AGAIN
2885
2886 033522                                    12$:
```

END OF SUB-PASS ROUTINE

```
                                .SBTTL  END OF SUB-PASS ROUTINE

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

033522  000004                  $EOSP:  SCOPE
033524  000240                          NOP
033526  013700  001466                  MOV     TSTQUE,R0       ;GET POINTER TO TSTQUE
033532  062700  000002                  ADD     #2,R0           ;ADJUST POINTER TO NEXT DEVICE
033536  010037  001466                  MOV     R0,TSTQUE       ;SAVE POINTER TO TSTQUE
033542  005710                          TST     (R0)            ;ANY MORE DEVICES FOR TEST ?
033544  001402                          BEQ     1$              ;BR IF NO
033546  000137  033776                  JMP     SHUT            ;YES, JUMP TO 'SHUT' ROUTINE
033552  012737  001470  001466  1$:     MOV     #TSTQUE+2,TSTQUE        ;INITIALIZE POINTER TO FIRST DEVICE IN
                                                                ;TEST QUE TABLE

                                .SBTTL  END OF PASS ROUTINE

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

033560                          $EOP:
033560  000240                          NOP
033562  005037  001116                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
033566  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
033572  005237  001230                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
033576  042737  100000  001230          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
033604  005327                          DEC     (PC)+           ;;LOOP?
033606  000001                  $EOPCT: .WORD   1
033610  003066                          BGT     $DOAGN          ;;YES
033612  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
033614  000001                  $ENDCT: .WORD   1
033616  033606                          $EOPCT
033620  104401  033626                  TYPE    ,65$            ;;TYPE ASCIZ STRING
033624  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
033644                          64$:
033644  013746  001230                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
033650  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
033652  005737  001126                  TST     $ERTTL          ;;SEE IF ANY ERRORS THIS PASS
033656  001431                          BEQ     $GT42P          ;;BR IF NO ERRORS TO REPORT
033660  104401  033666                  TYPE    ,67$            ;;TYPE ASCIZ STRING
033664  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
033730                          66$:
033730  013746  001126                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
033734  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
033736  005037  001126                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
```

END OF PASS ROUTINE

```
        033742  104401  001217          $GT42P: TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
        033746  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
        033752  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
        033754  000005                          RESET                   ;;CLEAR THE WORLD
        033756  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
        033760  000240                          NOP                     ;;SAVE ROOM
        033762  000240                          NOP                     ;;FOR
        033764  000240                          NOP                     ;;ACT11
        033766                          $DOAGN:
        033766  000137                          JMP     @(PC)+          ;;RETURN
        033770  033776                  $RTNAD: .WORD   SHUT
        033772     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
                                                .EVEN
    21
    22  033776  005737  001326          SHUT:   TST     CTLFG           ;WAS CONTROL C FLAGGED ?
    23  034002  001002                          BNE     1$              ;BR IF YES
    24  034004  000137  007524                  JMP     READY           ;CONTINUE
    25  034010  005737  000042          1$:     TST     @#42            ;ANY MONITOR PRESENT ?
    26  034014  001002                          BNE     2$              ;BR IF YES
    27  034016  000137  005434                  JMP     START           ;GO TO START
    28  034022  005037  001300          2$:     CLR     $DEVM           ;FUDGE NO DRIVES IN MAP
    29  034026  000137  033560                  JMP     $EOP            ;RETURN TO $EOP
```

```
                        .SBTTL  TEST PREPARATION MODULE

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

                        ;CALL:
                        ;       JSR     PC,TSTPRP
                        ;       .WORD   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 OR SKI ERROR
                        ;       BIT 10 = 1      RECALIBRATE DRIVE
                        ;       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

034032                  TSTPRP:

                        ;STORE TASK DESCRIPTOR AND CLEAR USER'S ERROR CALL
034032  017637  000000  034756          MOV     @(SP),39$       ;STORE DESCRIPTOR
034040  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
034044  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR CALL
034050  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN

034054  004737  040024                  JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
034060  004737  040110                  JSR     PC,GET          ;GET RMER2
034064  000411                          BR      2$              ;BR IF NO ERROR DETECTED
034066  000401                          BR      1$              ;GET OVER ERROR NUMBER
034070  000000                          .WORD   0               ;ERROR DEFINED BY GET SUBROUTINE
034072  062716  000004          1$:     ADD     #4,(SP)         ;XFER ERROR TO USER AND
034076  113776  034070  000000          MOVB    1$-2,@(SP)      ;GET ERROR NUMBER.
034104  000137  034746                  JMP     37$

034110  013737  001400  034760  2$:     MOV     RMER2I,40$      ;GET RMER2 AND SAVE FOR LATER

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

```
                                        ;SELECT DEVICE AND VERIFY DEVICE AVAILABLE IF BIT 15 SET IN TASK
034116  005737  034756                          TST     39$             ;SELECT DEVICE??
034122  100014                                  BPL     4$              ;NO!!

034124  004737  046336                          JSR     PC,DEVSEL       ;GO SELECT DEVICE
034130  000411                                  BR      4$              ;NO ERROR - CONTINUE
034132  000401                                  BR      3$
034134  000000                          .WORD   0                       ;ERROR NUMBER FROM DEVSEL
034136  062716  000004                  3$:     ADD     #4,(SP)         ;TRANSFER ERROR TO USER
034142  113776  034134  000000                MOVB    3$-2,@(SP)
034150  000137  034746                          JMP     37$

                                        ;;**************************************************************************
                                        ;CLEAR CONTROLLER IF BIT 14 IS SET IN TASK
034154                                  4$:
034154  032737  040000  034756                  BIT     #BIT14,39$      ;CLEAR CONTROLLER??
034162  001451                                  BEQ     13$             ;NO!!

034164  004737  050030                          JSR     PC,CNTCLR       ;GO CLEAR CONTROLLER
034170  000411                                  BR      7$              ;CONTINUE - NO ERROR
034172  000401                                  BR      6$
034174  000000                          5$:     .WORD   0               ;ERROR NUMBER FROM CNTCLR
034176  062716  000004                  6$:     ADD     #4,(SP)         ;TRANSFER ERROR TO USER
034202  113776  034174  000000                MOVB    5$,@(SP)
034210  000137  034746                          JMP     37$

                                        ;;**************************************************************************
                                        ;VERIFY CONTROLLER CLEAR IF BIT6 SET IN TASK
034214                                  7$:
034214  032737  000100  034756                  BIT     #BIT6,39$       ;VERIFY??
034222  001431                                  BEQ     13$             ;NO!!

034224  004737  040110                          JSR     PC,GET          ;GO GET STATUS
034230  000411                                  BR      10$             ;NO ERROR GETTING STATUS
034232  000401                                  BR      9$
034234  000000                          8$:     .WORD   0               ;ERROR FROM GETTING STATUS
034236  062716  000004                  9$:     ADD     #4,(SP)         ;TRANSFER ERROR TO USER
034242  113776  034234  000000                MOVB    8$,@(SP)
034250  000137  034746                          JMP     37$

034254  004737  050146                  10$:    JSR     PC,CLRSTS       ;GO VERIFY STATUS CLEAR
034260  000412                                  BR      13$             ;NO ERROR IN CLEAR
034262  000401                                  BR      12$
034264  000000                          11$:    .WORD   0               ;ERROR IN STATUS CLEAR
034266  005726                          12$:    TST     (SP)+           ;STRIP RETURN ADDRESS TO
034270  062716  000004                          ADD     #4,(SP)         ;SUBROUTINE AND TRANSFER
034274  113776  034264  000000                MOVB    11$,@(SP)       ;ERROR TO USER
034302  000137  034746                          JMP     37$

                                        ;;**************************************************************************
                                        ;EXECUTE PACK ACKNOWLEDGE IF BIT12 SET IN TASK AND VOLUME IS
                                        ;NOT VALID
034306                                  13$:
034306  032737  010000  034756                  BIT     #BIT12,39$      ;PACK ACKNOWLEDGE??
034314  001501                                  BEQ     25$             ;NO!!

034316  004737  040110                          JSR     PC,GET
```

```
                        BR      16$             ;NO ERROR GETTING RMDS
                        BR      15$
                14$:    .WORD   0
                15$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                        MOVB    14$,@(SP)
                        JMP     37$

                16$:    BIT     #VV,RMDSI       ;IS VOLUME VALID??
                        BNE     25$             ;YES!!

                        MOV     #PAKACK!GO,RMCS10       ;LOAD PACK ACK COMMAND
                        MOVB    #RMCS1,PUTINX   ;SETUP REGISTER INDEX TABLE
                        MOVB    #200,PUTINX+1
                        JSR     PC,PUT          ;GO WRITE COMMAND
                        BR      19$             ;NO ERROR LOADING REGISTER
                        BR      18$
                17$:    .WORD   0               ;ERROR FROM PUT SUB
                18$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                        MOVB    17$,@(SP)
                        BR      37$

                19$:    JSR     PC,TIMOUT       ;WAIT FOR COMMAND TO COMPLETE

                ;;*****************************************************************
                ;VERIFY PACK ACKNOWLEDGE IF #BIT4 SET IN TASK
                        BIT     #BIT4,39$       ;VERIFY PACK ACKNOWLEDGE??
                        BEQ     25$             ;NO!!

                        JSR     PC,GET          ;GO GET STATUS
                        BR      22$             ;NO ERROR GETTING STATUS
                        BR      21$
                20$:    .WORD   0               ;ERROR FROM GET SUB
                21$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                        MOVB    20$,@(SP)
                        BR      37$

                22$:    JSR     PC,ACKSTS       ;GO CHECK ACKNOWLEDGE
                        BR      25$             ;NO ERROR
                        BR      24$
                23$:    .WORD   0               ;PACK ACKNOWLEDGE ERROR
                24$:    TST     (SP)+           ;STRIP RETURN TO SUB AND
                        ADD     #4,(SP)         ;TRANSFER ERROR TO USER
                        MOVB    23$,@(SP)
                        BR      37$

                ;;*****************************************************************
                ;RECALIBRATE DRIVE IF BIT 11 IS SET IN TASK AND "SKI" IS SET
                ;OR "PIP" IS ACTIVE.
                ;RECALIBRATE DRIVE IF BIT 10 IS SET
                25$:
                        BIT     #BIT10,39$      ;RECALIBRATE DRIVE ?
                        BNE     29$             ;YES!!
                        BIT     #BIT11,39$      ;RECALIBRATE??
                        BEQ     38$             ;NO!!

                        JSR     PC,GET          ;GO GET RMDS
                        BR      28$             ;NO ERROR GETTING RMDS
```

```
                                        BR      27$
                        26$:            .WORD   0                       ;ERROR FROM GET SUB
                        27$:            ADD     #4,(SP)                 ;TRANSFER ERROR TO USER
                                        MOVB    26$,@(SP)
                                        BR      37$

                        28$:            BIT     #SKI,40$                ;WAS SKI SET ?
                                        BNE     29$                     ;YES, GO RECALIBRATE
                                        BIT     #PIP,RMDSI              ;IS PIP ACTIVE??
                                        BEQ     33$                     ;NO!!

                        29$:            MOV     #RECAL!GO,RMCS1         ;LOAD RECALIBRATE COMMAND
                                        MOVB    #RMCS1,PUTINX           ;AND REGISTER INDEX
                                        MOVB    #200,PUTINX+1           ;SET TERMINATOR
                                        JSR     PC,PUT                  ;GO ISSUE RECALIBRATE
                                        BR      31$                     ;NO ERROR
                                        BR      30$
                                        .WORD   0                       ;ERROR IN REGISTER TRANSFER
                        30$:            ADD     #4,(SP)                 ;TRANSFER ERROR TO USER
                                        MOVB    30$-2,@(SP)
                                        BR      37$

                        31$:            JSR     PC,TIMOUT               ;WAIT FOR COMPLETION

                        ;;**************************************************************
                        ;VERIFY RECALIBRATE IF BIT 3 SET IN TASK
                                        BIT     #BIT3,39$               ;VERIFY RECALIBRATE??
                                        BEQ     38$                     ;NO!!

                                        JSR     PC,GET                  ;GO GET STATUS
                                        BR      34$                     ;NO ERROR GETTING STATUS
                                        BR      33$
                        32$:            .WORD   0                       ;ERROR FROM GET
                        33$:            ADD     #4,(SP)                 ;TRANSFER ERROR TO USER
                                        MOVB    32$,@(SP)
                                        BR      37$

                        34$:            JSR     PC,RCLSTS               ;GO CHECK RECALIBRATE
                                        BR      38$                     ;NO ERROR DURING RECALIBRATE
                                        BR      36$
                        35$:            .WORD   0                       ;ERROR DURING RECALIBRAYE
                        36$:            TST     (SP)+                   ;STRIP RETURN TO SUB AND
                                        ADD     #4,(SP)                 ;TRANSFER ERROR TO USER
                                        MOVB    35$,@(SP)
                        37$:            SUB     #2,(SP)                 ;MOVE SP BACK BEFORE ERROR
                        38$:            NOP
                                        RTS     PC                      ;RETURN TO USER

                        39$:            .WORD   0                       ;TASK/VERIFY DESCRIPTOR
                        40$:            .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)  .RMDAD - THE TRACK AND SECTOR ADDRESS,
                        ;   (3)  .RMWCO - THE WORD COUNT,
                        ;   (4)  .RMCS10 - THE COMMAND,
                        ;   (5)  .RMOFO - THE FORMAT MODE AND SKIP SECTOR ERROR INHIBIT.
                        ;
                        ;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.
                        ;
034762  062716  000006  BADSCT: ADD     #6,(SP)         ;CLEAR ERROR NUMBER IN USER'S
034766  105076  000000          CLRB    @(SP)           ;ERROR CALL.
034772  162716  000006          SUB     #6,(SP)
                        ;
                        ;TEST "MEDIA ENABLE" TO DETERMINE WHETHER OR NOT THE BAD SECTOR FILES
                        ;HAVE BEEN RECOVERED.
034776  005737  001512          TST     MEDENB          ;HAVE BAD SECTOR FILES BEEN RECOVERED ?
035002  001402                  BEQ     1$              ;BR IF NO
035004  000137  036340          JMP     56$             ;YES, BAD SECTOR FILE IS AVAILABLE
                        ;
                        ;RECOVER SKIP SECTOR FILE FROM CYLINDER = 559., TRACK = 0 AND
                        ;RECOVER THE MANUFACTURES / USERS BAD SECTOR FILE FROM CYLINDER = 558.,
                        ;TRACK = LAST TRACK (RM80 = 13.). ALSO, SAVE THE USER'S PUT BUFFER
035010                  1$:
035010  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
035012  005000                  CLR     R0              ;START WITH RMCS1
035014  016060  001412  104254  2$:     MOV     PUTBUF(R0),BUFFER(R0)
035022  062700  000002          ADD     #2,R0           ;ADVANCE TO NEXT BUFFER POSITION
035026  022700  000046          CMP     #46,R0          ;END OF BUFFER
035032  103370                  BHIS    2$              ;NO !!
                        ;
                        ;SET RETRY COUNT AND LOAD PUT BUFFER AND REGISTER INDEX TABLE
                        ;SETUP PARAMETERS TO READ SKIP SECTOR FILE FIRST (3 SECTORS)
035034  012737  000003  037054          MOV     #3,74$          ;RETRY COUNT
035042  012737  001057  001446          MOV     #559.,RMDCO     ;DESIRED CYLINDER = 559.
```

BAD SECTOR MODULE

```
     57 035050  012737  000000  001420          MOV     #0,RMDAO        ;STARTING TRACK = 0, SECTOR = 0
     58 035056  012737  176372  001414          MOV     #-258.*3,RMWCO  ;258. * 3 SECTORS (2'S COMP)
     59 035064  012737  010000  001444          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
     60 035072  012737  105260  001416          MOV     #BUFTWO,RMBAO   ;SETUP BUS ADDRESS TO READ SKIP SECTOR FILE
     61
     62 035100  012700  001555                  MOV     #PUTINX,R0      ;R0 POINTS TO REGISTER INDEX TABLE
     63 035104  112720  000006                  MOVB    #RMDA,(R0)+
     64 035110  112720  000034                  MOVB    #RMDC,(R0)+
     65 035114  112720  000002                  MOVB    #RMWC,(R0)+
     66 035120  112720  000032                  MOVB    #RMOF,(R0)+
     67 035124  112720  000004                  MOVB    #RMBA,(R0)+
     68 035130  112720  000000                  MOVB    #RMCS1,(R0)+
     69 035134  112720  000200                  MOVB    #200,(R0)+
     70 035140  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     71
     72                                 ;SET GET INDEX TABLE FOR READING STATUS
     73 035142                          3$:
     74 035142  004737  040024                  JSR     PC,GETSTS       ;SETUP GET INDEX REGISTER FOR STATUS
     75 035146  004737  040110                  JSR     PC,GET          ;GET REGISTERS
     76 035152  000411                          BR      5$              ;BR IF NO ERROR
     77 035154  000401                          BR      4$              ;JUMP OVER ERROR NUMBER
     78 035156  000000                          .WORD   0               ;ERROR DEFINED BY GET SUB
     79 035160  062716  000006          4$:     ADD     #6,(SP)         ;XFER ERROR TO USER AND
     80 035164  113776  035156  000000          MOVB    4$-2,@(SP)      ;GET ERROR NUMBER.
     81 035172  000137  035766                  JMP     42$
     82
     83 035176  013737  001400  037066  5$:     MOV     RMER2I,79$      ;GET RMER2 AND SAVE FOR LATER
     84
     85                                 ;CLEAR THE DEVICE USING DRIVE CLEAR COMMAND
     86 035204  012737  000011  001412          MOV     #DRVCLR!GO,RMCS10       ;LOAD COMMAND IN PUT BUFFER
     87 035212  004737  040360                  JSR     PC,PUT          ;OUTPUT COMMAND
     88 035216  000411                          BR      8$              ;RETURN HERE IF NO ERROR
     89 035220  000401                          BR      7$              ;GET AROUND ERROR #
     90 035222  000000                  6$:     .WORD   0               ;ERROR # GOES HERE
     91 035224  062716  000006          7$:     ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
     92 035230  113776  035222  000000          MOVB    6$,@(SP)        ;MOVE ERROR NUMBER TO USER
     93 035236  000137  035766                  JMP     42$
     94
     95 035242  004737  040732          8$:     JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
     96 035246  004737  040110                  JSR     PC,GET          ;GO GET STATUS
     97 035252  000411                          BR      11$             ;RETURN HERE IF NO ERROR
     98 035254  000401                          BR      10$             ;GET AROUND ERROR #
     99 035256  000000                  9$:     .WORD   0               ;ERROR # GOES HERE
    100 035260  062716  000006          10$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
    101 035264  113776  035256  000000          MOVB    9$,@(SP)        ;MOVE ERROR # TO USERS ERROR CALL
    102 035272  000137  035766                  JMP     42$
    103
    104 035276  004737  053364          11$:    JSR     PC,DRVSTS       ;GO VERIFY DRIVE CLEAR COMMAND
    105 035302  000412                          BR      14$             ;RETURN HERE IF NO ERROR
    106 035304  000401                          BR      13$             ;GET AROUND ERROR
    107 035306  000000                  12$:    .WORD   0               ;ERROR # GOES HERE
    108 035310  005726                  13$:    TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
    109 035312  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
    110 035316  113776  035306  000000          MOVB    12$,@(SP)       ;MOVE ERROR # TO USER CALL
    111 035324  000137  035766                  JMP     42$
    112
    113                                 ;ISSUE A PACK ACKNOWLEDGE IF VOLUME VALID IS RESET
```

```
                                14$:
035330                          
035330  032737  000100  001350          BIT     #VV,RMDSI       ;IS VV RESET ??
035336  001052                          BNE     23$             ;NO !!
                                
035340  012737  000023  001412          MOV     #PACACK!GO,RMCS1O       ;LOAD COMMAND
035346  004737  040360                  JSR     PC,PUT          ;GO PUT COMMAND TO DRIVE
035352  000411                          BR      17$             ;RETURN HERE IF NO OUTPUT ERROR
035354  000401                          BR      16$             ;GET AROUND ERROR #
035356  000000                  15$:    .WORD   0               ;ERROR # GOES HERE
035360  062716  000006          16$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035364  113776  035356  000000          MOVB    15$,@(SP)       ;MOVE ERROR # TO ERROR CALL
035372  000137  035766                  JMP     42$
                                
035376  004737  040732          17$:    JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
035402  004737  040110                  JSR     PC,GET          ;GO GET STATUS FOR PACK ACK
035406  000411                          BR      20$             ;RETURN HERE IF NO ERROR
035410  000401                          BR      19$             ;GET AROUND ERROR #
035412  000000                  18$:    .WORD   0               ;ERROR # GOES HERE
035414  062716  000006          19$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035420  113776  035412  000000          MOVB    18$,@(SP)       ;MOVE ERROR # TO CALL
035426  000137  035766                  JMP     42$
                                
035432  004737  051026          20$:    JSR     PC,ACKSTS       ;GO VERIFY ACKNOWLEDGE STATUS
035436  000412                          BR      23$             ;RETURN HERE IF NO ERROR
035440  000401                          BR      22$             ;GET AROUND ERROR #
035442  000000                  21$:    .WORD   0               ;ERROR # GOES HERE
035444  005726                  22$:    TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
035446  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035452  113776  035442  000000          MOVB    21$,@(SP)       ;MOVE ERROR # TO USERS ERROR CALL
035460  000137  035766                  JMP     42$
                                
                                ;RECALIBRATE THE DRIVE IF "SKI" OR 'PIP IS SET
                                23$:
035464  032737  040000  037066          BIT     #SKI,79$        ;WAS SKI SET ?
035472  001004                          BNE     24$             ;YES, GO RECALIBRATE
035474  032737  020000  001350          BIT     #PIP,RMDSI      ;IS PIP SET ??
035502  001452                          BEQ     32$             ;NO !!
                                
035504  012737  000007  001412  24$:    MOV     #RECAL!GO,RMCS1O        ;LOAD RECALIBRATE COMMAND
035512  004737  040360                  JSR     PC,PUT          ;PUT THE RECAL COMMAND
035516  000411                          BR      26$             ;RETURN HERE IF NO ERROR
035520  000401                          BR      25$             ;GET AROUND ERROR #
035522  000000                          .WORD   0               ;ERROR # GOES HERE
035524  062716  000006          25$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035530  113776  035522  000000          MOVB    25$-2,@(SP)     ;MOVE ERROR # TO USERS CALL
035536  000137  035766                  JMP     42$
                                
035542  004737  040732          26$:    JSR     PC,TIMOUT       ;WAIT FOR RECALIBRATE TO COMPLETE
035546  004737  040110                  JSR     PC,GET          ;GO GET RECAL STATUS
035552  000411                          BR      29$             ;RETURN HERE IF NO ERROR
035554  000401                          BR      28$             ;GET AROUND ERROR #
035556  000000                  27$:    .WORD   0               ;ERROR # GOES HERE
035560  062716  000006          28$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
035564  113776  035556  000000          MOVB    27$,@(SP)       ;MOVE ERROR TO USERS CALL
035572  000137  035766                  JMP     42$
                                
035576  004737  051622          29$:    JSR     PC,RCLSTS       ;GO VERIFY RECALIBRATE STATUS
```

```
                                        BR      32$             ;RETURN HERE IF NO ERROR
                                        BR      31$             ;GET AROUND ERROR #
                                30$:    .WORD   0               ;ERROR # GOES HERE
                                31$:    TST     (SP)+           ;STRIP RETURN TO SUBROUTINE
                                        ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    30$,@(SP)       ;MOVE ERROR # TO USERS CALL
                                        JMP     42$

                                ;READ THE SECTOR IDENTIFIED BY RMDAD, INCLUDING HEADER AND DATA
                                32$:
                                        MOV     #RH!GO,RMCS10   ;LOAD READ HEADER AND DATA COMMAND
                                        JSR     PC,PUT          ;PUT COMMAND
                                        BR      35$             ;RETURN HERE IF NO ERROR
                                        BR      34$             ;GET AROUND ERROR #
                                33$:    .WORD   0               ;ERROR # GOES HERE
                                34$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    33$,@(SP)       ;MOVE ERROR # TO USERS ERROR CALL
                                        JMP     42$

                                35$:    JSR     PC,TIMOUT       ;WAIT FOR READ OPERATION TO COMPLETE
                                        JSR     PC,GET          ;GO GET STATUS FOR READ OPERATION
                                        BR      38$             ;RETURN HERE IF NO ERROR
                                        BR      37$             ;GET AROUND ERROR #
                                36$:    .WORD   0               ;ERROR # GOES HERE
                                37$:    ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    36$,@(SP)       ;MOVE ERROR # TO CALL
                                        JMP     42$

                                38$:    JSR     PC,DTASTS       ;GO VERIFY RESULTS OF READ OPERATION
                                        BR      41$             ;RETURN HERE IF NO ERROR
                                        BR      40$             ;GET AROUND ERROR #
                                39$:    .WORD   0               ;ERROR # GOES HERE
                                40$:    TST     (SP)+           ;STRIP RETURN ADDRESS TO SUBROUTINE
                                        ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    39$,@(SP)       ;MOVE ERROR # TO USERS CALL
                                        JMP     42$

                                41$:    BIT     #TRE,RMCS1I     ;ANY CONTROLLER ERRORS ?
                                        BNE     42$             ;BR IF YES
                                        BR      51$             ;NO ERRORS DETECTED

                                ;;*****************************************************************
                                ;AN ERROR HAS BEEN DETECTED IN TRYING TO READ THE BAD SECTOR FILE.
                                ;THE SECTOR WILL BE RETRIED IF POSSIBLE.

                                42$:    DEC     74$             ;YES, DECREMENT RETRY COUNT AND
                                        BNE     48$             ;RETRY IF COUNT IS POSITIVE

                                ;THE RETRY COUNT HAS EXPIRED - SEE IF THE ERROR IS MEDIA RELATED

                                        BIT     #DCK!HCRC!HCE!FER!ECH,RMER1I    ;ANY MEDIA RELATED ERRORS ?
                                        BNE     43$             ;YES, GO TRY NEXT AVAILABLE SECTOR

                                        BIT     #BSE!SSE,RMER2I ;ANY MEDIA RELATED ERRORS ?
                                        BEQ     49$             ;NO, EXIT AND REPORT ERROR ON RETURN

                                ;THE ERRORS DETECTED WHILE TRYING TO RECOVER THE BAD SECTOR FILE ARE
```

```
                                        ;DUE TO THE MEDIA.  SEE IF THE BAD SECTOR FILE CAN BE RECOVERED FROM
                                        ;ANOTHER AREA ON THE LAST TRACK

036014  032737  000040  001400  43$:    BIT     #SSE,RMER2I     ;SSE ERROR ?
036022  001406                          BEQ     44$             ;NO !!
036024  005237  001420                  INC     RMDAO           ;DISPLACE SECTOR ADDRESS BY 1 TO
                                                                ;ACCOUNT FOR THE SKIP SECTOR ERROR
036030  052737  001000  001444          BIS     #SSEI,RMOFO     ;AND SET SSEI IN OFFSET REGISTER.
036036  000452                          BR      47$             ;TRY AGAIN

036040  062737  000002  001420  44$:    ADD     #2,RMDAO        ;ADVANCE SECTOR ADDRESS BY 2
036046  022737  001056  001446          CMP     #558.,RMDCO     ;READING DEC 144 BAD SECTOR FILE ?
036054  001407                          BEQ     45$             ;YES !!
036056  005237  001420                  INC     RMDAO           ;IF READING SSF, THEN ADVANCE SECTOR ADDRESS BY 1 MORE
036062  122737  000017  001420          CMPB    #15.,RMDAO      ;QUIT IF ALL SSF SECTORS HAVE BEEN
036070  101424                          BLOS    49$             ;TRIED.
036072  000414                          BR      47$

036074  022737  106264  001416  45$:    CMP     #MFGFIL,RMBAO   ;READING MANUFACTURES BAD FILE ?
036102  001004                          BNE     46$             ;NO !!
036104  122737  000012  001420          CMPB    #10.,RMDAO      ;QUIT IF ALL MFG SECTORS HAVE BEEN
036112  101413                          BLOS    49$             ;TRIED.
036114  122737  000040  001420  46$:    CMPB    #32.,RMDAO      ;QUIT IF ALL USER SECTORS HAVE BEEN
036122  101407                          BLOS    49$             ;TRIED.

036124  012737  000003  037054  47$:    MOV     #3,74$          ;REINSTATE RETRY COUNT FOR THIS SECTOR
036132  162716  000006          48$:    SUB     #6,(SP)         ;MOVE SP BACK TO NO ERROR
036136  000137  035142                  JMP     3$              ;RETRY THE READ OPERATION

                                        ;THE BAD SECTOR FILE CANNOT BE READ
036142                          49$:
036142  000240                          NOP
036144  032777  020000  143002          BIT     #SW13,@SWR      ;INHIBIT MESSAGE ?
036152  001002                          BNE     50$             ;YES
036154  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO ERROR RETURN
036160  000137  037050          50$:    JMP     73$             ;GO TO MODULE EXIT

                                        ;THE SECTOR WAS RECOVERED WITHOUT ERROR - READ THE MFG/USER FILE IF
                                        ;THIS IS THE SSF FILE OR ELSE DONE.
036164                          51$:
036164  022737  001056  001446          CMP     #558.,RMDCO     ;READING DEC 144 FILES ?
036172  001422                          BEQ     52$             ;YES !
036174  004737  037070                  JSR     PC,GETSSF       ;GET SKIP SECTOR FILE FOR USE

036200  012737  001056  001446          MOV     #558.,RMDCO     ;READ DEC 144 FILE, CYLINDER 558.
036206  013737  001334  001420          MOV     LSTRK,RMDAO     ;MFG FILE LAST TRACK, SECTOR = 0
036214  012737  010000  001444          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
036222  012737  106264  001416          MOV     #MFGFIL,RMBAO   ;POINT TO MANUFACTURES FILE BUFFER
036230  012737  177376  001414          MOV     #-258.,RMWCO    ;2 + 256. WORDS (2'S COMP)
036236  000415                          BR      53$             ;GO READ MFG FILE
036240                          52$:
036240  022737  107272  001416          CMP     #USRFIL,RMBAO   ;WAS THE USER FILE READ ??
036246  001416                          BEQ     54$             ;YES - READ IS COMPLETE
036250  112737  000012  001420          MOVB    #10.,RMDAO      ;READ THE USER FILE LAST TRACK, SECTOR = 10.
036256  012737  107272  001416          MOV     #USRFIL,RMBAO   ;POINT TO USERS FILE BUFFER
036264  012737  010000  001444          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
```

```
    285 036272  012737  000003  037054  53$:    MOV     #3,74$          ;RELOAD THE RETRY COUNT FOR THIS SECTOR
    286 036300  000137  035142                  JMP     3$              ;GO READ THE USER FILE
    287
    288                                         ;SET MEDIA ENABLE AND RESTORE THE USERS PUT BUFFER
    289 036304                                  54$:
        036304  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
    290 036306  005000                          CLR     R0              ;R0 IS REGISTER INDEX
    291 036310  012737  177777  001512          MOV     #-1,MEDENB
    292 036316  016060  104254  001412  55$:    MOV     BUFFER(R0),PUTBUF(R0)
    293 036324  062700  000002                  ADD     #2,R0           ;ADVANCE R0
    294 036330  022700  000046                  CMP     #46,R0          ;DONE ??
    295 036334  103370                          BHIS    55$
    296 036336  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
    297
    298                                         ;;**************************************************************
    299                                         ;VERIFY THAT THE DESIRED SECTOR IS NOT IN THE MFG BAD SECTOR FILE,
    300                                         ;NOT IN THE USERS BAD SECTOR FILE AND NOT IN THE SSF BAD SECTOR FILE.
    301                                         ;ASSIGN A NEW SECTOR IF THE SECTOR IS IN ANY OF THE FILES.
    302                                         ;;**************************************************************
    303
    304                                         ;LOAD INITIAL VARIABLES AND COMPUTE THE NUMBER OF SECTORS
    305 036340                                  56$:
        036340  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
        036342  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
        036344  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
    306 036346  013737  001446  001516          MOV     RMDCO,ASNDC     ;LOAD REQUESTED CYLINDER, TRACK,
    307 036354  013737  001420  001520          MOV     RMDAO,ASNDA     ;AND SECTOR ADDRESS IN ASSIGNED STORAGE
    308 036362  005002                          CLR     R2              ;R2 = NUMBER OF SECTORS
    309 036364  013700  001414                  MOV     RMWCO,R0        ;R0 = WORD COUNT
    310 036370  005400                          NEG     R0              ;MAKE NUMBER POSITIVE
    311 036372  012701  000400                  MOV     #256.,R1        ;R1 = NUMBER OF WORDS PER SECTOR
    312 036376  032737  000002  001412          BIT     #BIT1,RMCS10    ;IS THIS A HEADER AND DATA COMMAND ??
    313 036404  001402                          BEQ     57$             ;NO !!
    314 036406  012701  000402                  MOV     #258.,R1        ;CHANGE WORDS PER SECTOR
    315 036412  020100                  57$:    CMP     R1,R0           ;IS THERE A FULL SECTOR ??
    316 036414  101404                          BLOS    58$             ;YES !!
    317 036416  005700                          TST     R0              ;IS R0 ZERO ??
    318 036420  001405                          BEQ     59$             ;YES !!
    319 036422  005202                          INC     R2              ;INCREMENT FOR PARTIAL SECTOR
    320 036424  000403                          BR      59$
    321 036426  160100                  58$:    SUB     R1,R0           ;SUBTRACT ONE SECTOR FROM WORD COUNT
    322 036430  005202                          INC     R2              ;INCREMENT SECTOR COUNT
    323 036432  000767                          BR      57$
    324 036434  010237  037054          59$:    MOV     R2,74$          ;SAVE SECTOR COUNT
    325
    326                                         ;LOAD PARAMETERS AND SEARCH THE MFG, USER AND THE MAPPED SSF BAD
    327                                         ;SECTOR FILES FOR THE ASSIGNED SECTOR. ALSO, SEARCH THE ADJACENT
    328                                         ;SECTORS IF THE SECTOR COUNT IS MORE THAN ONE.
    329
    330 036440  012737  106300  037064          MOV     #MFGFIL+14,78$  ;THE STARTING ADDRESS OF MFG FILE
    331                                                                 ;TO BASE ADDRESS STORAGE.
    332 036446  004737  036506                  JSR     PC,60$          ;GO SEARCH FILE
    333 036452  012737  107306  037064          MOV     #USRFIL+14,78$  ;LOAD STARTING ADDRESS OF USR FILE
    334                                                                 ;TO BASE ADDRESS STORAGE.
    335 036460  004737  036506                  JSR     PC,60$          ;GO SEARCH FILE
    336 036464  012737  110304  037064          MOV     #SSFIL+4,78$    ;LOAD STARTING ADDRESS OF MAPPED SSF
    337                                                                 ;FILE TO BASE ADDRESS STORAGE.
```

```
                                        CLR     SSFENB          ;ASSUME 'SSF' BIT IS NOT TO BE WRITTEN
                                                                ;IN THE 1ST HEADER WORD.
                                        JSR     PC,60$          ;SEARCH SKIP SECTOR MAPPED FILE
                                        JMP     71$             ;DONE WITH ALL FILE SEARCHES !!

                                60$:    MOV     ASNDC,76$       ;LOAD COMPARING CYLINDER ADDRESS
                                        MOV     ASNDA,77$       ;LOAD COMPARING TRACK, SECTOR ADDRESS
                                        MOV     74$,75$         ;LOAD NUMBER OF SECTORS TO CONFIRM

                                ;SETUP FOR A BINARY SEARCH OF THE CURRENT FILE FOR THE COMPARING
                                ;CYLINDER, TRACK AND SECTOR ADDRESS

                                61$:    MOV     78$,R0          ;LOAD THE BASE ADDRESS IN R0
                                62$:    CMP     #-1,(R0)        ;IS THIS FILE TERMINATOR ?
                                        BEQ     67$             ;BR IF YES
                                        CMP     76$,(R0)        ;IS THIS CYLINDER IN BAD TABLE ?
                                        BNE     63$             ;BR IF NO

                                ;FILE ENTRY EQUALS COMPARING CYLINDER.  SEE IF THE NEXT ENTRY EQUALS
                                ;THE COMPARING TRACK, AND SECTOR.

                                        CMPB    77$+1,3(R0)     ;IS THIS TRACK IN BAD TABLE ?
                                        BNE     63$             ;BR IF NO

                                        CMPB    77$,2(R0)       ;IS THIS SECTOR IN BAD TABLE ?
                                        BEQ     64$             ;BR IF YES
                                        BLT     63$             ;BR IF SECTOR IS LESS THAN TABLE ENTRY, ELSE
                                        MOV     #-1,SSFENB      ;ENABLE 'SSF' BIT TO BE WRITTEN IN THE
                                                                ;1ST HEADER WORD.

                                63$:    CMP     (R0)+,(R0)+     ;NO, ADJUST CYLINDER POINTER IN BAD FILE
                                        BR      62$             ;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.
                                64$:
                                        INCB    ASNDA           ;INCREMENT SECTOR
                                        BIT     #SSEI,RMOFO     ;IS SSEI SET ?
                                        BEQ     65$             ;NO !!
                                        CMPB    #31.,ASNDA      ;SECTOR OK ?
                                        BHIS    60$             ;YES !!
                                        BR      66$
                                65$:    CMPB    #30.,ASNDA      ;SECTOR OK ??
                                        BHIS    60$             ;YES !!
                                66$:    CLRB    ASNDA           ;CLEAR SECTOR AND ADVANCE TRACK
                                        INCB    ASNDA+1
                                        CMPB    LSTRK+1,ASNDA+1 ;TRACK OK ?
                                        BHIS    60$             ;YES !!
                                        CLR     ASNDA           ;CLEAR TRACK AND SECTOR
                                        INC     ASNDC           ;INCREMENT CYLINDER
                                        CMP     #560.,ASNDC     ;CYLINDER OK ??
                                        BHIS    60$             ;YES !!
                                        TST     (SP)+           ;RESTORE STACK AND
                                        BR      71$             ;GET OUT !!

                                ;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.
036706                          67$:
036706  005337  037056                DEC     75$             ;DECREMENT NUMBER OF SECTORS TO COMPARE
036712  001456                        BEQ     73$             ;DONE IF ZERO

036714  105237  037062                INCB    77$             ;INCREMENT THE COMPARING SECTOR
036720  032737  001000  001444        BIT     #SSEI,RMOFO     ;IS SSEI SET ?
036726  001405                        BEQ     68$             ;NO !!
036730  122737  000037  037062        CMPB    #31.,77$        ;SECTOR OK ?
036736  103032                        BHIS    70$             ;YES !!
036740  000404                        BR      69$
036742  122737  000036  037062  68$:  CMPB    #30.,77$        ;SECTOR OK ??
036750  103025                        BHIS    70$             ;YES !!
036752  105037  037062        69$:    CLRB    77$             ;CLEAR SECTOR
036756  105237  037063                INCB    77$+1           ;INCREMENT TRACK
036762  123737  001335  037063        CMPB    LSTRK+1,77$+1   ;TRACK OK ??
036770  103015                        BHIS    70$             ;YES !!
036772  005037  037062                CLR     77$             ;CLEAR SECTOR TRACK
036776  005237  037060                INC     76$             ;INCFEMENT CYLINDER
037002  022737  001060  037060        CMP     #560.,76$       ;CYLINDER OK ??
037010  103005                        BHIS    70$             ;YES !!
037012  013737  037060  001446        MOV     76$,RMDCO       ;LOAD CYLINDER
037020  005726                        TST     (SP)+           ;RESTORE STACK AND
037022  000407                        BR      72$             ;GET OUT !!

037024  000641                70$:    BR      61$             ;SEARCH NEXT SECTOR

                                        ;ASSIGN THE SECTOR AND RETURN TO USER
037026                          71$:
037026  013737  001516  001446        MOV     ASNDC,RMDCO     ;LOAD CYLINDER
037034  013737  001520  001420        MOV     ASNDA,RMDAO     ;LOAD TRACK AND SECTOR
037042                          72$:
037042  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
037044  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
037046  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0

037050  000240                73$:    NOP
037052  000207                        RTS     PC

                                        ;THE FOLLOWING ARE STORAGE LOCATIONS FOR THE MODULE

037054  000000                74$:    .WORD   0       ;RETRY COUNT/ NUMBER OF SECTORS REQUIRED
037056  000000                75$:    .WORD   0       ;NUMBER OF SECTORS TO COMPARE
037060  000000                76$:    .WORD   0       ;COMPARING CYLINDER
037062  000000                77$:    .WORD   0       ;COMPARING TRACK AND SECTOR
037064  000000                78$:    .WORD   0       ;BASE ADDRESS OF BAD SECTOR FILE BEING SEARCHED
037066  000000                79$:    .WORD   0       ;CONTAINS RMER2

                                        ;THIS ROUTINE SEARCHES THE SKIP SECTOR FILE FOR ANY BAD SECTORS
                                        ;WHICH OCCUR ON THE FE CYLINDERS. THE BAD SECTORS ARE THEN MAPPED
                                        ;INTO A 'SSFIL' TABLE TO BE USED BY THE 'BADSCT' ROUTINE.

037070                          GETSSF:
037070  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037072  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037074  012700  105260                MOV     #BUFTWO,R0      ;R0 = POINTER TO BUFFER
037100  012701  110300                MOV     #SSFIL,R1       ;R1 = POINTER TO SKIP SECTOR TABLE
```

BAD SECTOR MODULE

```
    447 037104  012021                        MOV     (R0)+,(R1)+     ;SAVE HEADER WORD 0(CYL ADDRESS)
    448 037106  012021                        MOV     (R0)+,(R1)+     ;SAVE HEADER WORD 1(TRK/SEC ADDRESS)
    449
    450 037110  012721  177777          1$:   MOV     #-1,(R1)+       ;INITIALIZE 'SSFIL' TABLE
    451 037114  022701  110374                CMP     #SSFIL+60.,R1   ;DONE ?
    452 037120  001373                        BNE     1$              ;BR IF NO
    453
    454 037122  012701  110304                MOV     #SSFIL+4,R1     ;SETUP TO MAP SSFIL TABLE
    455 037126  062700  000026                ADD     #22.,R0         ;GET OVER S/N(2), CREATE DATE, REV DATE, REV#,
    456                                                               ;FORMAT TYPE, UNUSED WORDS(3), ENTRY# AND
    457                                                               ;CHECKSUM (11. WORDS)
    458 037132  021027  001057          2$:   CMP     (R0),#559.      ;IS CYLINDER IN FE AREA ?
    459 037136  103410                        BLO     4$              ;NO !!
    460 037140  101004                        BHI     3$              ;YES, ON CYLINDER 560.
    461                                                               ;YES, WHEN ON CYLINDER 559. DO NOT LOG
    462                                                               ;ANY BAD SECTORS THAT OCCUR ON THE SKIP
    463                                                               ;SECTOR TRACK(0) OR ALTERNATE DEC144 TRACK(1)
    464 037142  126027  000003  000001        CMPB    3(R0),#1        ;IS IT ON TRACK 0 OR 1 ?
    465 037150  003403                        BLE     4$              ;YES !!
    466 037152  011021                  3$:   MOV     (R0),(R1)+      ;LOG THIS CYLINDER AND
    467 037154  016021  000002                MOV     2(R0),(R1)+     ;TRACK/SECTOR IN 'SSFIL' TABLE.
    468
    469 037160  022020                  4$:   CMP     (R0)+,(R0)+     ;ADJUST BUFFER TO NEXT CYLINDER ENTRY
    470 037162  020127  110374                CMP     R1,#SSFIL+60.   ;ANY MORE ENTRIES AVAILABLE IN TABLE ?
    471 037166  103003                        BHIS    5$              ;NO, EXIT !!
    472 037170  020027  110274                CMP     R0,#BUFTWO+1548.        ;DONE WITH ALL 3 SECTORS ?
    473 037174  103756                        BLO     2$              ;NO, DO NEXT ENTRY
    474
    475 037176  012737  177777  107270  5$:   MOV     #-1,MFGFIL+516. ;RESTORE TERMINATOR TO MFGFIL
    476 037204  000240                        NOP
    477 037206  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
        037210  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
    478 037212  000207                        RTS     PC
```

```
                                .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

 037214                         GENBUF:
 037214  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 037216  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 037220  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 037222  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 037224  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 037226  013700  001416                 MOV     RMBA0,R0        ;LOAD DATA BUFFER ADDRESS
 037232  013701  001414                 MOV     RMWC0,R1        ;LOAD WORD COUNT
 037236  013737  001446  037456         MOV     RMDC0,7$        ;LOAD STARTING CYLINDER ADDRESS
 037244  013737  001420  037460         MOV     RMDA0,8$        ;LOAD STARTING TRACK,SECTOR ADDRESS

 037252  032737  000002  001412  1$:    BIT     #BIT1,RMCS10    ;WRITE HEADER & DATA??
 037260  001451                         BEQ     3$              ;NO!!
 037262  013710  037456                 MOV     7$,(R0) ;WRITE HEADER WORD #1
 037266  052710  140000                 BIS     #MSE!USE,(R0)   ;SET BAD SECTOR FLAGS FOR GOOD SECTOR
 037272  012702  000035                 MOV     #29.,R2         ;R2 = MAXIMUM SECTOR ADDRESS (29.)

 037276  032737  010000  001444         BIT     #FMT16,RMOF0    ;18 BIT FORMAT??
 037304  001410                         BEQ     2$              ;YES !!
 037306  052710  010000                 BIS     #FMT16,(R0)     ;SET 16 FORMAT BIT IN HEADER
 037312  005202                         INC     R2              ;CHANGE MAXIMUM SECTOR ADDRESS (30.)

 037314  032737  001000  001444         BIT     #SSEI,RMOF0     ;IS SSEI SET TO ACCESS ALL SECTORS ?
 037322  001401                         BEQ     2$              ;NO !!
 037324  005202                         INC     R2              ;CHANGE MAXIMUM SECTOR ADDRESS (31.)

 037326  005201                 2$:     INC     R1              ;INCREMENT WORD COUNT
 037330  001443                         BEQ     6$              ;EXIT IF DONE

 037332  005720                         TST     (R0)+           ;MOVE R0 TO HEADER WORD #2
 037334  013720  037460                 MOV     8$,(R0)+        ;WRITE HEADER WORD #2
 037340  005201                         INC     R1              ;INCREMENT WORD COUNT AND
 037342  001436                         BEQ     6$              ;EXIT IF DONE
 037344  012703  037460                 MOV     #8$,R3          ;ADVANCE SECTOR ADDRESS
 037350  105213                         INCB    (R3)
 037352  120213                         CMPB    R2,(R3)         ;SECTOR OVERFLOW ??
 037354  103013                         BHIS    3$              ;NO !!
 037356  105013                         CLRB    (R3)            ;YES - CLEAR SECTOR ADDRESS
```

BUFFER GENERATOR SUBROUTINE

```
 037360  105263  000001                  INCB    1(R3)               ;ADVANCE TRACK ADDRESS
 037364  123763  001335  000001          CMPB    LSTRK+1,1(R3)       ;TRACK OVERFLOW ??
 037372  103004                          BHIS    3$                  ;NO !!
 037374  105063  000001                  CLRB    1(R3)               ;YES - CLEAR TRACK ADDRESS
 037400  105237  037456                  INCB    7$                  ;ADVANCE CYLINDER ADDRESS
 037404  012704  000400          3$:     MOV     #256.,R4            ;LOAD SECTOR DATA COUNT
 037410  013702  001174          4$:     MOV     $TMP0,R2            ;LOAD PATTERN ADDRESS
 037414  013703  001176                  MOV     $TMP1,R3            ;LOAD PATTERN COUNT
 037420  012220                  5$:     MOV     (R2)+,(R0)+         ;WRITE DATA PATTERN
 037422  005201                          INC     R1                  ;INCREMENT WORD COUNT AND
 037424  001405                          BEQ     6$                  ;EXIT IF DONE
 037426  005304                          DEC     R4                  ;DECREMENT SECTOR COUNT
 037430  001710                          BEQ     1$                  ;START NEXT SECTOR IF 0
 037432  005303                          DEC     R3                  ;DECREMENT PATTERN COUNT
 037434  001765                          BEQ     4$                  ;RESTART PATTERN IF 0
 037436  000770                          BR      5$                  ;CONTINUE DATA PATTERN
 037440                          6$:
 037440  012604                          MOV     (SP)+,R4            ;;POP STACK INTO R4
 037442  012603                          MOV     (SP)+,R3            ;;POP STACK INTO R3
 037444  012602                          MOV     (SP)+,R2            ;;POP STACK INTO R2
 037446  012601                          MOV     (SP)+,R1            ;;POP STACK INTO R1
 037450  012600                          MOV     (SP)+,R0            ;;POP STACK INTO R0
 037452  000240                          NOP
 037454  000207                          RTS     PC

 037456  000000          7$:     .WORD                               ;CYLINDER ADDRESS STORAGE
 037460  000000          8$:     .WORD                               ;TRACK, SECTOR ADDRESS STORAGE
```

COMPARE BUFFER SUBROUTINE

```
                                .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)    ???
                                
037462                          CMPBUF:
037462  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037464  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037466  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037470  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
037472  005037  040022                 CLR     14$             ;CLEAR CORRECTION FLAG

                                ;DETERMINE IF DATA SHOULD BE CORRECTED
037476  033737  004000  001370         BIT     ECI,RMOFI       ;WAS ECC CORRECTION ALLOWED ??
037504  001063                         BNE     9$              ;NO !!
037506  032737  100000  001352         BIT     #DCK,RMER1I     ;WAS THERE A DATA CHECK ??
037514  001457                         BEQ     9$              ;NO !!
037516  032737  000100  001352         BIT     #ECH,RMER1I     ;IS ERROR CORRECTION HARD SET ?
037524  001053                         BNE     9$              ;YES !!
037526  032737  010000  001370         BIT     #FMT16,RMOFI    ;IS THIS 16 BIT FORMAT ??
037534  001447                         BEQ     9$              ;NO !!

                                ;CORRECT DATA USING ECC INFORMATION
037536  013700  001416                 MOV     RMBAO,R0        ;R0 = STARTING BUFFER ADDRESS
037542  013701  001402                 MOV     RMEC1I,R1       ;R1 = ECC POSITION
037546  052737  100000  040022         BIS     #BIT15,14$      ;SET CORRECTION FLAG

                                ;MOVE R0 TO WORD BOUNDARY OF ERROR BURST
037554  022701  000020         1$:     CMP     #16.,R1         ;IS BIT POSITION > 1 WORD
037560  103004                         BHIS    2$              ;NO !!
037562  162701  000020                 SUB     #16.,R1         ;SUBTRACT 1 WORDS WORTH
037566  005720                         TST     (R0)+           ;ADVANCE BUFFER ADDRESS 1 WORD
037570  000771                         BR      1$
037572  012702  000001         2$:     MOV     #1,R2           ;R2 = BIT POINTER
037576  010203                         MOV     R2,R3           ;R3 = BIT NUMBER

                                ;MOVE R2 TO STARTING BIT OF ERROR BURST
037600  020301                 3$:     CMP     R3,R1           ;IS R3 SAME AS R1 ??
037602  001403                         BEQ     4$              ;YES !!
037604  006302                         ASL     R2              ;SHIFT BIT POINTER
037606  005203                         INC     R3              ;INCREMENT BIT NUMBER
037610  000773                         BR      3$
037612  012703  000013         4$:     MOV     #11.,R3         ;R3 = LENGTH OF ERROR BURST

                                ;CORRECT THE ERROR BURST
037616  030237  001404         5$:     BIT     R2,RMEC2I       ;IS THIS BIT SET IN ECC PATTERN ??
```

COMPARE BUFFER SUBROUTINE

```
                                        BEQ     7$              ;NO - DO NOT CORRECT THIS BIT
                                        BIT     R2,(R0)         ;IS THE BIT PRESENTLY SET ??
                                        BEQ     6$              ;NO
                                        BIC     R2,(R0)         ;RESET THE BIT
                                        BR      7$
                                6$:     BIS     R2,(R0)         ;SET THE BIT
                                7$:     ASL     R2              ;SHIFT TO NEXT BIT
                                        BNE     8$
                                        MOV     #1,R2           ;CONTINUE WITH FIRST BIT OF NEXT WORD
                                        TST     (R0)+
                                8$:     DEC     R3              ;END OF BURST ??
                                        BNE     5$              ;NO !!

                                ;COMPARE WRITE BUFFER TO READ BUFFER
                                9$:     MOV     @10(SP),R0      ;R0 = WRITE BUFFER
                                        ADD     #2,10(SP)       ;MOVE SP TO READ ADDRESS
                                        MOV     @10(SP),R1      ;R1 = READ BUFFER
                                        ADD     #2,10(SP)       ;MOVE SP TO RETURN ADDRESS
                                        MOV     RMWCI,R2        ;R2 = NUMBER OF WORDS TRANSFER
                                        SUB     RMWCO,R2
                                10$:    CMP     (R0)+,(R1)+     ;COMPARE DATA WORDS
                                        BNE     11$             ;EXIT IF NOT EQUAL
                                        DEC     R2              ;DECREMENT WORD COUNT
                                        BNE     10$             ;CONTINUE IF NOT DONE
                                        BR      13$             ;DONE COMPARE - NO ERROR

                                ;DATA COMPARE FAILED
                                11$:    MOV     -(R0),$GDDAT    ;STORE GOOD DATA FOR TYPEOUT
                                        MOV     -(R1),$BDDAT    ;STORE BAD DATA FOR TYPEOUT
                                        MOV     R0,$GDADR       ;STORE ADDRESS OF GOOD DATA
                                        MOV     R1,$BDADR       ;STORE ADDRESS OF BAD DATA
                                        MOV     R2,$TMP0        ;STORE WORD COUNT OF ERROR
                                        ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #336,@10(SP)    ;WRITE ERROR NUMBER IN CALL

                                ;CHANGE ERROR NUMBER IF ECC CORRECTION FAILED
                                        BIT     #BIT15,14$      ;WAS ECC CORRECTION USED ??
                                        BEQ     12$             ;NO !!
                                        MOVB    #163,@10(SP)    ;ECC CORRECTION FAILED
                                12$:    SUB     #2,10(SP)       ;MOVE SP TO RETURN IF ERROR
                                        NOP
                                13$:
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        RTS     PC              ;RETURN TO USER

                                14$:    .WORD                   ;ECC CORRECTION FLAG
```

```
                        .SBTTL  GET STATUS SUBROUTINE
                        ;
                        ;THIS SUBROUTINE SETS UP THE "GET INDEX TABLE" AND THE "GET
                        ;BUFFER" FOR READING ALL SUBSYSTEM REGISTERS VIA THE GET SUBROUTINE
                        ;AND THEN RETURNS TO THE USER.
                        ;
                        ;CALL:  JSR     PC,GETSTS
                        ;       ???                             RETURN HERE
                        ;
040024                  GETSTS:
040024  010046                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040026  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040030  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040032  012700  001526         MOV     #GETINX,R0      ;R0 = ADDRESS OF INDEX TABLE
040036  012701  001406         MOV     #RMEC2I+2,R1    ;R1 = ADDRESS OF GET BUFFER
040042  012702  000046         MOV     #RMEC2,R2       ;R2 = REGISTER INDE
040046  110220         1$:     MOVB    R2,(R0)+        ;WRITE REGISTER INDEX IN TABLE
040050  005041                 CLR     -(R1)           ;CLEAR CORPESPONDING LOCATION
040052  162702  000002 2$:     SUB     #2,R2           ;DECREMENT TO NEXT INDEX
040056  100405                 BMI     3$              ;BRANCH OUT IF DONE
040060  022702  000022         CMP     #RMDB,R2        ;DONT WRITE RMDB INDEX
040064  001370                 BNE     1$
040066  005041                 CLR     -(R1)
040070  000770                 BR      2$
040072  112720  000200 3$:     MOVB    #200,(R0)+      ;WRITE TERMINATOR
040076  012602                 MOV     (SP)+,R2        ;;POP STACK INTO R2
040100  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
040102  012600                 MOV     (SP)+,R0        ;;POP STACK INTO R0
040104  000240                 NOP
040106  000207                 RTS     PC
```

GET SUBROUTINE

```
                                .SBTTL  GET SUBROUTINE
                                ;
                                ;THIS SUBROUTINE READS THE REGISTERS WHICH ARE LISTED IN THE
                                ;"GET INDEX TABLE" AND STORES THEIR VALUES IN THE CORRESPONDING
                                ;LOCATION IN THE "GET REGISTER BUFFER".  FOR EXAMPLE, AN
                                ;ENTRY OF 04 IN THE TABLE WILL CAUSE THE SUBROUTINE TO
                                ;READ "RMBA" AND STORE ITS CONTENTS AT THE LOCATION IN
                                ;THE BUFFER ASSIGNED TO THAT REGISTER.  THE NUMBER OF
                                ;REGISTERS TO BE READ IS VARIABLE FROM 1 TO 22; THE INDEX
                                ;TABLE MUST BE TERMINATED WITH A CONTROL BYTE (200)
                                ;WHICH SHOULD FOLLOW THE LAST ENTRY.
                                ;
                                ;SUBROUTINE CALL:
                                ;(1)     "GET INDEX TABLE" HAS BEEN LOADED WITH REGISTER INDEX
                                ;       VALUES AND TERMINATED WITH A CONTROL BYTE
                                ;(2)     "GET INPUT BUFFER" IS AVAILABLE FOR USE.  (NOTE THAT
                                ;       UNUSED LOCATIONS, I.E.,ENTRIES IN BUFFER CORRESPONDING
                                ;       TO REGISTERS NOT READ, ARE NOT CHANGED.)
                                ;(3)     JSR     PC,GET
                                ;       BR      ???             RETURN HERE IF NO ERROR FOUND
                                ;       NOP                     RETURN HERE IF ANY ERROR FOUND
                                ;       ERROR                   SUB DEFINES ERROR NUMBER
                                ;       ???
                                ;
                                ;R0 = REGISTER BASE ADDRESS
                                ;R1 = REGISTER ADDRESS
                                ;R2 = BUFFER BASE ADDRESS
                                ;R3 = BUFFER ADDRESS
                                ;R4 = POINTER TO REGISTER INDEX
                                ;
040110  000240                  GET:    NOP
040112  062716  000004                  ADD     #4,(SP)         ;CLEAR ERROR NUMBER IN USER'S
040116  105076  000000                  CLRB    @(SP)           ;ERROR CALL
040122  162716  000004                  SUB     #4,(SP)
040126  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040130  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040132  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040134  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
040136  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
040140  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
040144  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
040150  013700  001276                  MOV     $BASE,R0
040154  012702  001336                  MOV     #GETBUF,R2
040160  012704  001526                  MOV     #GETINX,R4
040164  012737  040272  000004          MOV     #3$,ERRVEC      ;SETUP FOR TIMEOUT
040172  012737  000300  000006          MOV     #PR6,ERRVEC+2
040200  016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
040206  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
040214  032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
040222  001007                          BNE     2$              ;YES!!
040224  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN USER'S
040232  112776  000112  000016          MOVB    #112,@16(SP)    ;ERROR CALL
040240  000423                          BR      4$
040242  105714                  2$:     TSTB    (R4)            ;DONE??
040244  100433                          BMI     6$              ;YES!!
040246  111401                          MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
040250  042701  177700                  BIC     #^CIDXMSK,R1    ;CLEAR ANY SIGN EXTENSION
```

GET SUBROUTINE

```
                                                ADD     R0,R1
                                                MOVB    (R4)+,R3        ;R3 = STORAGE ADDRESS FOR REGISTER
                                                BIC     #^CIDXMSK,R3    ;CLEAR ANY SIGN EXTENSION
                                                ADD     R2,R3
                                                MOV     (R1),(R3)       ;READ REGISTER
                                                BR      2$

                                        3$:     CMP     (SP)+,(SP)+     ;RESTORE STACK
                                                ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
                                                MOVB    #7,@16(SP)      ;USER'S ERROR CALL
                                        4$:     SUB     #2,16(SP)
                                        5$:     TSTB    (R4)            ;DONE CLEARING??
                                                BMI     6$              ;YES!!
                                                CLR     R3              ;CLEAR REMAINING STORAGE
                                                MOVB    (R4)+,R3        ;LOCATIONS
                                                ADD     R2,R3
                                                CLR     (R3)
                                                BR      5$
                                        6$:
                                                MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
                                                MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
                                                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
                                                RTS     PC              ;RETURN
```

PUT SUBROUTINE

```
                                .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)     "PL  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

040360  000240                  PUT:    NOP
040362  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040364  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040366  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040370  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
040372  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
040374  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
040400  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
040404  013700  001276                  MOV     $BASE,R0
040410  012702  001412                  MOV     #PUTBUF,R2
040414  012704  001555                  MOV     #PUTINX,R4
040420  012737  040540  000004          MOV     #4$,ERRVEC      ;SETUP FOR TIMEOUT
040426  012737  000300  000006          MOV     #PR6,ERRVEC+2
040434  016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
040442  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
040450  032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
040456  001007                          BNE     2$              ;YES!!
040460  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
040466  112776  000112  000016          MOVB    #112,@16(SP)    ;USER'S ERROR CALL
040474  000430                          BR      5$
040476  105714                  2$:     TSTB    (R4)            ;DONE??
040500  100431                          BMI     6$              ;YES!!
040502  111401                          MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
040504  042701  177700                  BIC     #^CIDXMSK,R1    ;CLEAR ANY SIGN EXTENSION
040510  060001                          ADD     R0,R1
040512  111403                          MOVB    (R4),R3         ;R3 = STORAGE ADDRESS
040514  042703  177700                  BIC     #^CIDXMSK,R3    ;CLEAR ANY SIGN EXTENSION
040520  060203                          ADD     R2,R3
040522  011311                          MOV     (R3),(R1)       ;WRITE REGISTER
040524  122714  000032                  CMPB    #RMOF,(R4)      ;WAS RMOF JUST LOADED ?
```

PUT SUBROUTINE

```
     52 040530  001001                          BNE     3$              ;BR IF NO
     53 040532  011311                          MOV     (R3),(R1)       ;WRITE RMOF AGAIN, TO ENSURE THAT 16 BIT MODE
     54                                                                 ;IS SET BEFORE "SSEI" BIT IS SET.
     55 040534  105724                  3$:     TSTB    (R4)+           ;ADJUST REGISTER POINTER
     56 040536  000757                          BR      2$
     57
     58 040540  022626                  4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
     59 040542  062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     60 040550  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     61 040556  162766  000002  000016  5$:     SUB     #2,16(SP)
     62
     63 040564                          6$:
        040564  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        040570  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        040574  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        040576  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        040600  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        040602  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        040604  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     64 040606  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL  SIZE CLOCK SUBROUTINE

                        SIZCLK:
040610 012746 000004            MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
040614 013746 000006            MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
040620 012737 040656 000004     MOV     #1$,ERRVEC      ;SET UP FOR BUS TIMEOUT
040626 012737 000300 000006     MOV     #PR6,ERRVEC+2
040634 012737 177546 001522     MOV     #177546,CLKADR  ;LOAD ADDRESSES FOR KW11-L
040642 012737 000100 001524     MOV     #100,CLKVCT
040650 005777 140646            TST     @CLKADR         ;TEST FOR KW11-L PRESENT
040654 000421                   BR      3$              ;YES - KW11-L IS PRESENT
040656 022626                   CMP     (SP)+,(SP)+     ;RESTORE SP
040660 012737 040710 000004     MOV     #2$,ERRVEC      ;SET UP FOR BUS TIMEOUT
040666 012737 172540 001522     MOV     #172540,CLKADR  ;LOAD ADDRESSES FOR KW11-P CLOCK
040674 012737 000104 001524     MOV     #104,CLKVCT
040702 005777 140614            TST     @CLKADR         ;TEST FOR KW11-P PRESENT
040706 000404                   BR      3$              ;YES - KW11-P IS PRESENT
040710 022626                   CMP     (SP)+,(SP)+     ;RESTORE SP
040712 062766 000002 000004     ADD     #2,4(SP)        ;MOVE RETURN TO ERROR
040720                  3$:
040720 012637 000006            MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
040724 012637 000004            MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
040730 000207                   RTS     PC              ;RETURN TO USER
```

```
                                .SBTTL  TIMEOUT SUBROUTINE
                                ;
                                ;THIS SUBROUTINE WAITS FOR RDY = 1 AND GO = 0 OR FOR A TIMEOUT
                                ;GREATER THAN APPROX. 500 MSEC AND THEN RETURNS.
                                ;
                                ;CALL:  JSR     PC,TIMOUT
                                ;       ???                             RETURN HERE
                                ;
040732                          TIMOUT:
040732  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040734  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040736  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
040742  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
040746  012737  041032  000004          MOV     #3$,ERRVEC      ;SETUP FOR BUS TIMEOUT - 04 TRAP
040754  012737  000300  000006          MOV     #PR6,ERRVEC+2
040762  013700  001276                  MOV     $BASE,R0        ;R0=BASE ADDRESS
040766  012702  000036                  MOV     #30.,R2         ;R2=NUMBER OF CLOCK CYCLES
040772  004737  041074          1$:     JSR     PC,STIMER       ;START CLOCK TIMER
                                ;
040776  016046  000000          2$:     MOV     RMCS1(R0),-(SP) ;GET STATUS
041002  042716  177576                  BIC     #^C<RDY!GO>,(SP)
041006  022726  000200                  CMP     #RDY,(SP)+      ;RDY=1,GO=0??
041012  001421                          BEQ     4$              ;YES!!
041014  032777  000200  140500          BIT     #BIT7,@CLKADR   ;TIMER DONE??
041022  001765                          BEQ     2$              ;NO!!
041024  005302                          DEC     R2              ;DEC NUMBER OF CYCLES
041026  001361                          BNE     1$              ;CONTINUE IF NOT DONE
041030  000412                          BR      4$              ;"RDY" DID NOT SET OR "GO" DID NOT RESET
                                                                ;WITHIN 500 MSEC AFTER THE COMMAND WAS ISSUED.
041032  022626          3$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
041034  062766  000004  000012          ADD     #4,12(SP)       ;MOVE SP TO USER'S CALL
041042  112776  000007  000012          MOVB    #7,@12(SP)      ;WRITE ERROR NUMBER
041050  162766  000002  000012          SUB     #2,12(SP)
                                ;
041056                          4$:
041056  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
041062  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
041066  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
041070  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
041072  000207                          RTS     PC              ;RETURN TO USER
                                ;
                                ;THIS ROUTINE IS USED START THE KW11-L OR THE KW11-P CLOCK. THE CLOCK
                                ;IS STARTED TO RUN IN FLAG MODE. A CLOCK WILL BE 16.667MS FOR A 60 HZ CPU
                                ;AND 20MS FOR A 50 HZ CPU.
                                ;
041074                          STIMER:
041074  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041076  013701  001522                  MOV     CLKADR,R1       ;R1=CLOCK ADDRESS
041102  020127  172540                  CMP     R1,#172540      ;KW11-P CLOCK??
041106  001003                          BNE     1$              ;NO!!
041110  012761  000001  000002          MOV     #1,2(R1)        ;SET COUNTER
041116  012711  000005          1$:     MOV     #BIT2!BIT0,(R1) ;START COUNTER
041122  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
041124  000207                          RTS     PC              ;RETURN
```

```
                        .SBTTL  PRIMARY ERROR CHECK SUBROUTINE

                        ;THE PURPOSE OF THIS SUBROUTINE IS TO VERIFY THAT STATUS IS VALID AND
                        ;THAT FURTHER ERROR AND STATUS CHECKING SHOULD BE PERFORMED.  THE
                        ;FOLLOWING CHECKS ARE MADE:
                        ;
                        ;       .CORRECT UNIT IS SELECTED, I.E., THE UNIT SELECT BITS OF RMCS2
                        ;(BITS 0-2) EQUAL THE UNIT BEING TESTED;
                        ;
                        ;       .SELECTED UNIT IS AVAILABLE, I.E., DVA (BIT 11 OF RMCS1) IS SET
                        ;AND NED (BIT 12 OF RMCS2) IS RESET;
                        ;
                        ;       .LAST COMMAND WAS COMPLETED, I.E., THE MASSBUS CONTROLLER IS
                        ;READY (BIT 7 OF RMCS1) AND THE GO BIT IS RESET (BIT 0 OF RMCS1) OR THE
                        ;DRIVE READY BIT (BIT 7 OF RMDS) IS SET.
                        ;       .NO PARITY ERROR OCCURRED WHEN READING REMOTE REGISTERS,
                        ;I.E., MCPE = 0.
                        ;       .NO PARITY ERROR OCCURRED WHEN WRITING REMOTE REGISTERS,
                        ;I.E., PAR = 0, OR, PAR = DPE = 1
                        ;
                        ;THE SUBROUTINE ASSUMES THAT:
                        ;
                        ;       .STATUS HAS BEEN STORED IN THE REGISTER INPUT BUFFER,
                        ;IN PARTICULAR, RMCS1, RMCS2 AND RMDS HAVE BEEN STORED IN THEIR
                        ;CORRESPONDING LOCATIONS OF THE "GET" BUFFER.
                        ;
                        ;       .($UNIT) CONTAINS THE DRIVE NUMBER
                        ;
                        ;THE SUBROUTINE IS CALLED AS FOLLOWS:
                        ;
                        ;(1)    JSR     PC,PRIERR
                        ;       BR      ???             RETURN HERE IF NO ERROR
                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                        ;       ???                     RETURN HERE IF NO MORE ERRORS

041126                  PRIERR:

                        ;CLEAR USER'S ERROR CALL
041126  062716  000004          ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
041132  105076  000000          CLRB    @(SP)           ;CLEAR ERROR NUMBER
041136  162716  000004          SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

                        ;REPORT AN ERROR IF THE WRONG UNIT IS SELECTED
041142  013737  001346  001142  MOV     RMCS2I,$BDDAT   ;CORRECT UNIT SELECTED??
041150  042737  177770  001142  BIC     #^CUNTMSK,$BDDAT
041156  013737  001234  001140  MOV     $UNIT,$GDDAT    ;GOOD DATA FOR TYPEOUT
041164  042737  177770  001140  BIC     #^CUNTMSK,$GDDAT
041172  123737  001140  001142  CMPB    $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                                        ;DRIVE NUMBERS
041200  001415                  BEQ     1$              ;YES!!
041202  062716  000004          ADD     #4,(SP)
041206  112776  000001  000000  MOVB    #1,@(SP)        ;ERROR 1
041214  162716  000002          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
041220  004736                  JSR     PC,@(SP)+       ;REPORT WRONG UNIT SELECTED
```

```
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP
                                        JMP     10$             ;SKIP OTHER CHECKS
                                1$:

                                ;REPORT AN ERROR IF THE DEVICE IS NOT AVAILABLE OR IF
                                ;THE DEVICE IS NONEXISTANT
                                        BIT     #DVA,RMCS1I     ;DEVICE AVAILABLE??
                                        BNE     3$              ;YES!!
                                        MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
                                        BIS     #DVA,$GDDAT
                                        MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)
                                        MOVB    #2,@(SP)        ;ERROR #2
                                        BIT     #NED,RMCS2I     ;WAS NED SET??
                                        BEQ     2$              ;NO!!
                                        MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
                                        MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
                                        BIC     #NED,$GDDAT
                                        MOVB    #3,@(SP)        ;YES - CHANGE ERROR NUMBER
                                2$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT DEVICE NOT AVAILABLE
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP
                                        BR      10$             ;SKIP OTHER CHECKS
                                3$:

                                ;REPORT AN ERROR IF MASSBUS CONTROLLER IS NOT READY
                                        BIT     #RDY,RMCS1I     ;CONTROLLER READY??
                                        BNE     4$              ;YES!!
                                        MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
                                        BIS     #RDY,$GDDAT
                                        BIC     #SC!TRE!MCPE!GO,$GDDAT
                                        MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)
                                        MOVB    #4,@(SP)        ;ERROR #4
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT CONTROLLER NOT READY
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP
                                        BR      10$             ;SKIP OTHER CHECKS
                                4$:

                                ;REPORT AN ERROR IF GO IS NOT ZERO AND DRY IS NOT ONE
                                        BIT     #GO,RMCS1I      ;GO RESET??
                                        BEQ     5$              ;YES!!
                                        BIT     #DRY,RMDSI      ;DRIVE READY??
                                        BNE     5$              ;YES!!
                                        MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #SC!TRE!MCPE!GO,$GDDAT
                                        MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)
                                        MOVB    #5,@(SP)        ;ERROR #5
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT DRIVE NOT READY
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP
```

```
                                        BR      10$
                                5$:
                                ;
                                ;REPORT AN ERROR IF THE RH CONTROLLER DETECTED BAD
                                ;PARITY ON THE MASSBUS CONTROL BUS
001336                                  BIT     #MCPE,RMCS1I    ;PARITY ERROR ??
                                        BEQ     6$              ;NO!!
001336  001140                          MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
001336  001140                          BIC     #SC!TRE!MCPE!GO,$GDDAT
001336  001142                          MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
000013  000000                          MOVB    #13,@(SP)       ;ERROR #13
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                        SUB     #10,(SP)        ;RESTORE STACK
                                        NOP
                                        BR      10$
                                6$:
                                ;
                                ;REPORT AN ERROR IF THE RM80 DETECTED A CONTROL BUS PARITY ERROR
001352                                  BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
                                        BEQ     11$             ;NO!!
001400                                  BIT     #DPE,RMER2I     ;WAS IT THE CONTROL BUS??
                                        BNE     11$             ;NOT SURE!!
001426                                  BIT     #PAR,RMER1O     ;DID TEST SET PAR ??
                                        BEQ     9$              ;NO!!
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
001555                                  MOV     #PUTINX,R0      ;R0 POINTS TO INDEX TABLE
000014                          7$:     CMPB    #RMER1,(R0)     ;SEARCH TABLE FOR RMER1
                                        BNE     8$
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        BR      11$             ;PAR WAS SET BY TEST
                                8$:     TSTB    (R0)+           ;END OF TABLE??
                                        BPL     7$              ;NO!!
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
001352  001140                  9$:     MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
000010  001140                          BIC     #PAR,$GDDAT
001352  001142                          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
000004                                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
000050  000000                          MOVB    #50,@(SP)       ;WRITE THE ERROR NUMBER
000002                                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR
000010                                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
000010                          10$:    ADD     #10,(SP)        ;RETURN TO ERROR
                                11$:    NOP                     ;RETURN TO NO ERROR
                                        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 TH
                                ;INPUT REGISTER BUFFER.

041760                          SECERR:

                                ;STORE FUNCTION CODE AND CLEAR USER'S ERROR NUMBER
041760  013737  001412  046120          MOV     RMCS10,$1$      ;STORE FUNCTION CODE
041766  042737  177701  046120          BIC     #^C<F0!F1!F2!F3!F4>,$1$
041774  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
042000  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
042004  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

                                ;REPORT ERROR IF DRIVE IS NOT READY, I.E., IF DRY = 0
042010  032737  000200  001350          BIT     #DRY,RMDSI      ;DRIVE READY??
042016  001024                  BNE     1$              ;YES!!
042020  013737  001350  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
042026  042737  177577  001142          BIC     #^CDRY,$BDDAT
042034  012737  000200  001140          MOV     #DRY,$GDDAT     ;GOOD DATA FOR TYPEOUT
042042  062716  000004                  ADD     #4,(SP)
042046  112776  000010  000000          MOVB    #10,@(SP)       ;ERROR NUMBER
042054  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
042060  004736                  JSR     PC,@(SP)+       ;REPORT NOT READY
042062  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP) TO ERROR N
042066  000240                  NOP

                                ;REPORT ERROR IF GO BIT IS NOT RESET
042070  032737  000001  001336  1$:     BIT     #GO,RMCS1I      ;GO BIT RESET??
042076  001423                  BEQ     2$              ;YES!!
042100  013737  001336  001142          MOV     RMCS1I,$BDDAT   ;BAD DATA FOR TYPEOUT
042106  042737  177776  001142          BIC     #^CGO,$BDDAT
042114  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
042120  062716  000004                  ADD     #4,(SP)
042124  112776  000011  000000          MOVB    #11,@(SP)       ;ERROR NUMBER
042132  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
042136  004736                  JSR     PC,@(SP)+       ;REPORT DEVICE NOT AVAILABLE
042140  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
042144  000240                  NOP
```

SECONDARY ERROR CHECK SUBROUTINE

```
                                        ;REPORT ERROR IF FUNCTION CODE READ FROM DEVICE IS NOT CORRECT
042146  013737  001336  001142  2$:     MOV     RMCS1I,$BDDAT   ;IS FUNCTION CODE CORRECT??
042154  042737  177701  001142          BIC     #^C76,$BDDAT
042162  013737  046120  001140          MOV     81$,$GDDAT      ;EXPECTED FUNCTION CODE
042170  023737  001142  001140          CMP     $BDDAT,$GDDAT
042176  001413                          BEQ     3$              ;YES!!
042200  062716  000004                  ADD     #4,(SP)
042204  112776  000012  000000          MOVB    #12,@(SP)       ;ERROR NUMBER
042212  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
042216  004736                          JSR     PC,@(SP)+       ;REPORT WRONG FUNCTION CODE
042220  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
042224  000240                          NOP
042226                          5$:
                                        ;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
042226  005037  001140                  CLR     $GDDAT          ;EXPECT "ERR" = 0
042232  005737  001352                  TST     RMER1I          ;IS RMER1 = 0??
042236  001003                          BNE     4$              ;NO!!
042240  005737  001400                  TST     RMER2I          ;IS RMER2 = 0??
042244  001403                          BEQ     5$              ;YES!!
042246  052737  040000  001140  4$:     BIS     #ERR,$GDDAT     ;"ERR" SOULD BE SET
042254  013737  001350  001142  5$:     MOV     RMDSI,$BDDAT
042262  042737  137777  001142          BIC     #^CERR,$BDDAT
042270  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS "ERR" OK??
042276  001412                          BEQ     6$              ;YES!!
042300  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
042304  112776  000047  000000          MOVB    #47,@(SP)       ;WRITE ERROR NUMBER
042312  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
042316  004736                          JSR     PC,@(SP)+       ;REPORT INVALID COMP ERROR
042320  162716  000010                  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
042324  005037  001140          6$:     CLR     $GDDAT          ;EXPECT "TRE" = 0
042330  013746  001346                  MOV     RMCS2I,-(SP)    ;WAS DLT, WCE, UPE, NED, NEM
042334  042726  000377                  BIC     #377,(SP)+      ;PGE, MXF OR MDPE SET
042340  001010                          BNE     7$              ;YES!!
042342  032737  040000  001350          BIT     #ERR,RMDSI      ;WAS EXCEPTION RECEIVED??
042350  001407                          BEQ     8$              ;NO!!
042352  022737  000030  046120          CMP     #SEARCH,81$     ;WAS DATA TRANSFERRED??
042360  103003                          BHIS    8$              ;NO!!
042362  052737  040000  001140  7$:     BIS     #TRE,$GDDAT     ;"TRE" SHOULD BE SET
042370  013737  001336  001142  8$:     MOV     RMCS1I,$BDDAT   ;BAD DATA FOR TYPEOUT
042376  042737  137777  001142          BIC     #^CTRE,$BDDAT
042404  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS "TRE" OK??
042412  001413                          BEQ     9$              ;YES!!
042414  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
042420  112776  000014  000000          MOVB    #14,@(SP)       ;WRITE ERROR NUMBER
042426  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
042432  004736                          JSR     PC,@(SP)+       ;REPORT TRE ERROR
042434  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
042440  000240                          NOP
042442                          9$:
```

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

                                        ;GET AND STORE THE ENTRY FROM THE FUNCTION CODE TABLE
042442  010046                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042444  013700  046120                        MOV     81$,R0          ;GET FUNCTION CODE
042450  016037  067244  046112                MOV     FNCDTB(R0),78$  ;STORE ENTRY
042456  012600                                MOV     (SP)+,R0        ;;POP STACK INTO R0

                                        ;REPORT AN ERROR IF AN UNEXPECTED ATTENTION OCCURRED OR IF
                                        ;ATA IS NOT SET AND SHOULD BE SET.
042460  013737  046112  001140                MOV     78$,$GDDAT      ;GET EXPECTED ATA STATUS
042466  032737  040000  001350                BIT     #ERR,RMDSI      ;IS COMPOSITE ERROR SET ??
042474  001403                                BEQ     10$             ;NO !!
042476  052737  100000  001140                BIS     #ATA,$GDDAT     ;EXPECT AN ATTENTION
042504  042737  077777  001140  10$:          BIC     #^CATA,$GDDAT   ;STRIP DONT CARES
042512  013737  001350  001142                MOV     RMDSI,$BDDAT    ;GET RECEIVED ATA
042520  042737  077777  001142                BIC     #^CATA,$BDDAT   ;STRIP DONT CARES
042526  023737  001140  001142                CMP     $GDDAT,$BDDAT   ;IS ATA OK ??
042534  001413                                BEQ     11$             ;YES !!
042536  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
042542  112776  000006  000000                MOVB    #6,@(SP)        ;LOAD ERROR # IN CALL
042550  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
042554  004736                                JSR     PC,@(SP)+       ;REPORT ERROR
042556  162716  000010                        SUB     #10,(SP)        ;RESTORE SP
042562  000240                                NOP
042564                                  11$:

                                        ;REPORT ERROR IF ILF IS INCORRECT, I.E., IF ILF DOES NOT COMPARE
                                        ;WITH FUNCTION CODE TABLE
042564  013737  046112  001140                MOV     78$,$GDDAT      ;GET EXPECTED ILF
042572  042737  177776  001140                BIC     #^CILF,$GDDAT   ;CLEAR ALL OTHER BITS
042600  013737  001352  001142                MOV     RMER1I,$BDDAT   ;GET RECEIVED ILF
042606  042737  177776  001142                BIC     #^CILF,$BDDAT   ;CLEAR ALL OTHER BITS
042614  023737  001140  001142                CMP     $GDDAT,$BDDAT   ;IS ILF OK ??
042622  001412                                BEQ     12$             ;YES !!
042624  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
042630  112776  000254  000000                MOVB    #254,@(SP)      ;WRITE ERROR NUMBER IN CALL
042636  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
042642  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
042644  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
042650  005037  001140            12$:        CLR     $GDDAT          ;CLEAR EXPECTED STATUS

                                        ;REPORT AN ERROR IF WCE IS SET AND SHOULD NOT BE SET
042654  013746  046112                        MOV     78$,-(SP)       ;GET WCE STATUS ENABLE
042660  052716  137777                        BIS     #^CWCE,(SP)     ;SET ALL OTHER BITS
042664  013737  001346  001142                MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
042672  042637  001142                        BIC     (SP)+,$BDDAT    ;CLEAR WCE IF ENABLED
042676  001412                                BEQ     13$             ;BRANCH IF WCE OK
042700  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
042704  112776  000026  000000                MOVB    #26,@(SP)       ;WRITE ERROR NUMBER
042712  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
```

SECONDARY ERROR CHECK SUBROUTINE

```
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE ERROR
                                        13$:

                                        ;REPORT ERROR IF OPI STATUS IS SET AND SHOULD NOT BE SET
                                                MOV     78$,-(SP)       ;GET OPI STATUS ENABLE
                                                BIS     #^COPI,(SP)     ;SET ALL OTHER BITS
                                                MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR OPI IF ENABLED
                                                BEQ     14$             ;BRANCH IF OPI OK
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #164,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE SP
                                        14$:

                                        ;REPORT ERROR IF IVC IS SET AND IS NOT ENABLED OR IF IVC IS
                                        ;SET AND VV IS NOT RESET
                                                MOV     78$,-(SP)       ;GET IVC STATUS ENABLE
                                                BIT     #VV,RMDSI       ;IS VV SET
                                                BEQ     15$             ;NO !!
                                                BIC     #IVC,(SP)       ;YES - IVC SHOULD BE 0
                                        15$:    BIS     #^CIVC,(SP)     ;SET ALL OTHER BITS
                                                MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR IVC IF ENABLED
                                                BEQ     16$             ;BRANCH IF IVC OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #165,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        16$:

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

                                        ;REPORT AN ERROR IF WLE IS SET AND WRITE ERRORS ARE NOT ENABLED, OR IF
                                        ;THE DRIVE IS NOT WRITE PROTECTED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
                                                BIT     #WLE,78$        ;ARE WRITE ERRORS ENABLED ??
                                                BEQ     17$             ;NO !!
                                                BIT     #WRL,RMDSI      ;IS THE DRIVE WRITE PROTECTED ??
                                                BNE     18$             ;YES !!
                                        17$:    BIC     #WLE,(SP)       ;RESET WLE ENABLE
                                        18$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR WLE IF ENABLED
                                                BEQ     19$             ;BRANCH IF WLE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #23,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
```

```
    229 043144  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    230 043150                              19$:
    231
    232                                     ;REPORT ERROR IF WCF IS SET AND WRITE ERRORS ARE NOT ENABLED
    233 043150  012746  177777                      MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
    234 043154  032737  004000  046112              BIT     #WLE,78$        ;ARE WRITE ERRORS ENABLED ??
    235 043162  001002                              BNE     20$             ;YES !!
    236 043164  042716  000040                      BIC     #WCF,(SP)       ;DISABLE WCF ERROR
    237 043170  013737  001352  001142  20$:       MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
    238 043176  042637  001142                      BIC     (SP)+,$BDDAT    ;RESET WCF IF ENABLED
    239 043202  001412                              BEQ     21$             ;BRANCH IF WCF OK
    240 043204  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    241 043210  112776  000025  000000              MOVB    #25,@(SP)       ;WRITE ERROR NUMBER IN CALL
    242 043216  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    243 043222  004736                              JSR     PC,@(SP)+       ;REPORT ERROR
    244 043224  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    245 043230                              21$:
    246
    247                                     ;REPORT ERROR IF DPE IS SET AND WRITE ERRORS ARE NOT ENABLED
    248 043230  012746  177777                      MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
    249 043234  032737  004000  046112              BIT     #WLE,78$        ;ARE WRITE ERRORS ENABLED ??
    250 043242  001002                              BNE     22$             ;YES !!
    251 043244  042716  000010                      BIC     #DPE,(SP)       ;RESET DPE ENABLE
    252 043250  013737  001400  001142  22$:       MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
    253 043256  042637  001142                      BIC     (SP)+,$BDDAT    ;RESET DPE IF ENABLED
    254 043262  001412                              BEQ     23$             ;BRANCH IF DPE OK
    255 043264  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    256 043270  112776  000040  000000              MOVB    #40,@(SP)       ;WRITE ERROR NUMBER IN CALL
    257 043276  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    258 043302  004736                              JSR     PC,@(SP)+       ;REPORT ERROR
    259 043304  162716  000010                      SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    260 043310                              23$:
    261
    262                                     ;REPORT AN ERROR IF UPE IS SET AND WRITE ERRORS ARE NOT ENABLED
    263 043310  012746  177777                      MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
    264 043314  032737  004000  046112              BIT     #WLE,78$        ;ARE WRITE ERRORS ENABLED ??
    265 043322  001002                              BNE     24$             ;YES !!
    266 043324  042716  020000                      BIC     #UPE,(SP)       ;DISABLE UPE ERROR
    267 043330  013737  001346  001142  24$:       MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
    268 043336  042637  001142                      BIC     (SP)+,$BDDAT    ;RESET UPE IF ENABLED
    269 043342  001412                              BEQ     25$             ;BRANCH IF UPE OK
    270 043344  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    271 043350  112776  000024  000000              MOVB    #24,@(SP)       ;WRITE ERROR NUMBER IN CALL
    272 043356  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
    273 043362  004736                              JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    274 043364  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO NO ERROR
    275 043370                              25$:
    276
    277                                     ;REPORT AN ERROR IF IAE IS SET AND IS NOT ENABLED
    278 043370  013746  046112                      MOV     78$,-(SP)       ;GET IAE ENABLE
    279 043374  052716  175777                      BIS     #^CIAE,(SP)     ;SET ALL OTHER BITS
    280 043400  013737  001352  001142              MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
    281 043406  042637  001142                      BIC     (SP)+,$BDDAT    ;CLEAR IAE IF ENABLED
    282 043412  001412                              BEQ     26$             ;BRANCH IF IAE IS OK
    283 043414  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
    284 043420  112776  000166  000000              MOVB    #166,@(SP)      ;WRITE ERROR NUMBER
    285 043426  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
```

```
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                26$:

                                ;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
 043440  012746  177777                 MOV     #-1,-(SP)       ;ASSUME ERRORS ARE ENABLED
 043444  032737  001000  046112         BIT     #AOE,78$        ;ARE ERRORS ENABLED ??
 043452  001002                         BNE     27$             ;YES !!
 043454  042716  100000                 BIC     #DLT,(SP)       ;RESET DLT ENABLE
 043460  013737  001346  001142 27$:    MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
 043466  042637  001142                 BIC     (SP)+,$BDDAT    ;CLEAR DLT IF ENABLED
 043472  001412                         BEQ     28$             ;BRANCH IF DLT IS OK
 043474  062716  000004                 ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 043500  112776  000032  0C0000         MOVB    #32,@(SP)       ;WRITE ERROR NUMBER IN CALL
 043506  162716  000002                 SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
 043512  004736                         JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 043514  162716  000010                 SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 043520                          28$:

                                ;REPORT ERROR IF NEM IS SET AND READ/WRITE ERRORS ARE NOT ENABLED
 043520  012746  177777                 MOV     #-1,-(SP)       ;ASSUME ERRORS ARE ENABLED
 043524  032737  001000  046112         BIT     #AOE,78$        ;ARE ERRORS ENABLED ??
 043532  001002                         BNE     29$             ;YES !!
 043534  042716  004000                 BIC     #NEM,(SP)       ;DISABLE NEM
 043540  013737  001346  001142 29$:    MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
 043546  042637  001142                 BIC     (SP)+,$BDDAT    ;CLEAR NEM IF ENABLED
 043552  001412                         BEQ     30$             ;BRANCH IF NEM IS OK
 043554  062716  000004                 ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
 043560  112776  000167  000000         MOVB    #167,@(SP)      ;WRITE ERROR NUMBER IN CALL
 043566  162716  000002                 SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
 043572  004736                         JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 043574  162716  000010                 SUB     #10,(SP)        ;MOVE SP TO NO ERROR
 043600                          30$:

                                ;REPORT ERROR IF MXF IS SET AND READ/WRITE ERRORS ARE NOT ENABLED
 043600  012746  177777                 MOV     #-1,-(SP)       ;ASSUME ERRORS ARE ENABLED
 043604  032737  001000  046112         BIT     #AOE,78$        ;ARE DATA ERRORS ENABLED ??
 043612  001002                         BNE     31$             ;YES !!
 043614  042716  001000                 BIC     #MXF,(SP)       ;DISABLE MXF ERROR
 043620  013737  001346  001142 31$:    MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
 043626  042637  001142                 BIC     (SP)+,$BDDAT    ;CLEAR MXF IF ENABLED
 043632  001412                         BEQ     32$             ;BRANCH IF MXF IS OK
```

SECONDARY ERROR CHECK SUBROUTINE

```
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #33,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        32$:

                                        ;REPORT ERROR IF AOE IS SET AND DATA ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME DATA ERRORS ARE ENABLED
                                                BIT     #AOE,78$        ;ARE DATA ERRORS ENABLED ??
                                                BEQ     33$             ;NO !!
                                                BIT     #LBT,RMDSI      ;IS LBT ALSO SET ??
                                                BNE     34$             ;YES !!
                                        33$:    BIC     #AOE,(SP)       ;DISABLE AOE
                                        34$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR AOE IF ENABLED
                                                BEQ     35$             ;BRANCH IF AOE IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #20,@(SP)       ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        35$:

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

                                        ;RESET THE ENABLING BIT (HCE) IF HEADER COMPARE INHIBIT IS SET
                                                BIT     #HCI,RMOFI      ;IS HCI SET ??
                                                BEQ     36$             ;NO !!
                                                BIC     #HCE,78$        ;YES - DISABLE ALL HEADER ERRORS
                                        36$:

                                        ;REPORT AN ERROR IF HCRC IS SET AND HEADER ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                                BIT     #HCE,78$        ;ARE HEADER ERRORS ENABLED ??
                                                BNE     37$             ;YES !!
                                                BIC     #HCRC,(SP)      ;DISABLE HCRC
                                        37$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET HCRC IF ENABLED
                                                BEQ     38$             ;BRANCH IF HCRC IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #35,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        38$:

                                        ;REPORT ERROR IF HCE IS SET AND HEADER ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                                BIT     #HCE,78$        ;ARE ERRORS ENABLED ??
                                                BNE     39$             ;YES !!
                                                BIC     #HCE,(SP)       ;DISABLE HCE
                                        39$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR HCE IF ENABLED
```

```
                                        BEQ     40$             ;BRANCH IF HCE IS OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #36,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                40$:

                                ;REPORT ERROR IF FER IS SET AND HEADER ERRORS ARE NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME FER IS ENABLED
                                        BIT     #HCE,78$        ;ARE HEADER ERRORS ENABLED ??
                                        BNE     41$             ;YES !!
                                        BIC     #FER,(SP)       ;DISABLE FER
                                41$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;RESET FER IF ENABLED
                                        BEQ     42$             ;BRANCH IF FER OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #37,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                42$:

                                ;REPORT ERROR IF BSE IS SET AND HEADER ERRORS ARE NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                        BIT     #HCE,78$        ;ARE THEY ENABLED ??
                                        BNE     43$             ;YES !!
                                        BIC     #BSE,(SP)       ;DISABLE BSE
                                43$:    MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR BSE IF ENABLED
                                        BEQ     44$             ;BRANCH IF BSE OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #354,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                44$:

                                ;HCE IS USED AS THE ENABLING BIT FOR DETECTING SSE, SO REESTABLISH ENTRY
                                ;FROM THE FUNCTION CODE TABLE, TO CHECK THE "SSE" BIT
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     81$,R0          ;GET FUNCTION CODE
                                        MOV     FNCDTB(R0),78$  ;STORE FUNCTION TABLE ENTRY
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0

                                ;IF SKIP SECTOR ERROR INHIBIT (SSEI) IS SET OR FORMAT 16 (FMT) IS CLEAR,
                                ;RESET THE ENABLING BIT (HCE), OTHERWISE LEAVE ENABLING BIT (HCE) SET.
                                        BIT     #FMT16,RMOFI    ;18 BIT FORMAT MODE ?
                                        BEQ     45$             ;BR IF YES
                                        BIT     #SSEI,RMOFI     ;IS SSEI SET ?
                                        BEQ     46$             ;BR IF NO
                                45$:    BIC     #HCE,78$        ;DISABLE HEADER ERROR
                                46$:

                                ;REPORT ERROR IF SSE IS SET AND HEADER ERROR IS NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                        BIT     #HCE,78$        ;IS HCE ENABLED ?
```

SECONDARY ERROR CHECK SUBROUTINE

```
                                                BNE     47$             ;YES !!
                                                BIC     #SSE,(SP)       ;DISABLE SSE
                                        001142  47$:     MOV     RMER2I,$BDDAT   ;GET RECIEVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;CLEAR SSE IF NOT VALID FOR FUNCTION
                                                BEQ     48$             ;BRANCH IF SSE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000  MOVB    #63,@(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
                                        48$:

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

                                        ;REPORT ERROR IF MDPE IS SET AND IS NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                        046112  BIT     #ECH,78$        ;ARE DATA FIELD ERRORS ENABLED ??
                                                BNE     49$             ;YES !!
                                                BIC     #MDPE,(SP)      ;DISBALE MDPE
                                        001142  49$:     MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;CLEAR MDPE IF ENABLED
                                                BEQ     50$             ;BRANCH IF MDPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000  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
                                        50$:

                                        ;REPORT ERROR IF DCK IS SET AND DATA FIELD ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                        046112  BIT     #ECH,78$        ;ARE THEY ENABLED ??
                                                BNE     51$             ;YES !!
                                                BIC     #DCK,(SP)       ;DISABLE DCK
                                        001142  51$:     MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        001142          BIC     (SP)+,$BDDAT    ;CLEAR DCK IF ENABLED
                                                BEQ     52$             ;BRANCH IF DCK IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000  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
                                        52$:

                                        ;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
                                        046112  BIT     #ECH,78$        ;ARE ERRORS ENABLED ??
                                                BEQ     53$             ;NO !!
```

```
                                        BIT     #ECI,RMOFI      ;IS ECI SET ??
                                        BNE     53$             ;YES !!
                                        BIT     #DCK,RMER1I     ;IS DCK ALSO SET ??
                                        BNE     54$             ;YES !!
                                53$:    BIC     #ECH,(SP)       ;DISABLE ECH
                                54$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR ECH IF ENABLED
                                        BEQ     55$             ;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
                                55$:

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

                                        CMP     #SEARCH,81$     ;WAS DATA TRANSFERRED ?
                                        BLO     56$             ;BR IF YES
                                        JMP     75$             ;NO - EXIT

                                ;REPORT ERROR IF RMWC NOT ZERO AND TRE IS ZERO
                                56$:    MOV     RMWCI,$BDDAT    ;WORD COUNT ZERO??
                                        BEQ     57$             ;YES
                                        BIT     #TRE,RMCS1I     ;TRANSFER ERROR DETECTED??
                                        BNE     57$             ;YES!!
                                        ADD     #4,(SP)
                                        MOVB    #15,@(SP)       ;ERROR NUMBER
                                        CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT WORD COUNT NOT ZERO
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP

                                ;REPORT ERROR IF RMBA IS NOT CORRECT
                                57$:    MOV     RMWCI,$GDDAT    ;GET WORD COUNT AT END OF TRANSFER AND
                                        SUB     RMWCO,$GDDAT    ;SUBTRACT STARTING WORD COUNT.
                                        ASL     $GDDAT          ;* 2
                                        ADD     RMBAO,$GDDAT    ;ADD STARTING BUS ADDRESS

                                        BIT     #BAI,RMCS2I     ;WAS BUS ADDRESS INHIBIT (BAI) SET ??
                                        BEQ     58$             ;NO !!
                                        MOV     RMBAO,$GDDAT    ;ADDRESS SHOULD NOT HAVE CHANGED

                                58$:    CMP     $GDDAT,RMBAI    ;BUS ADDRESS OK??
                                        BEQ     59$             ;YES!!
                                        MOV     RMBAI,$BDDAT    ;BAD DATA FOR TYPEOUT
                                        ADD     #4,(SP)
                                        MOVB    #16,@(SP)       ;ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT UNEXPECTED ADDRESS
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP

                                ;COMPUTE NUMBER OF SECTORS TRANSFERRED FROM WORD COUNT
```

```
                                59$:    CLR     -(SP)           ;NUMBER OF SECTORS TRANSFERRED
                                        MOV     RMWCI,-(SP)     ;GET WORD COUNT AT END OF TRANSFER AND
                                        SUB     RMWCO,(SP)      ;SUBTRACT STARTING WORD COUNT.

                                        MOV     #256.,-(SP)     ;ASSUME 256. WORDS PER SECTOR
                                        BIT     #BIT1,RMCS10    ;HEADER & DATA COMMAND ??
                                        BEQ     60$             ;NO !!
                                        ADD     #2,(SP)         ;CHANGE TO 258. WORDS PER SECTOR

                                60$:    INC     4(SP)           ;INCREMENT SECTOR COUNT
                                        SUB     (SP),2(SP)      ;SUBTRACT ONE SECTOR'S WORTH
                                        BGT     60$             ;CONTINUE IF NOT DONE
                                        CMP     (SP)+,(SP)+     ;RESTORE STACK

                                ;COMPUTE EXPECTED SECTOR, TRACK AND CYLINDER ADDRESS FROM
                                ;NUMBER OF SECTORS
                                        MOV     RMDCO,78$       ;STORE ORIGINAL CYLINDER
                                        MOV     RMDAO,79$       ;STORE ORIGINAL TRACK
                                        MOV     RMDAO,80$       ;STORE ORIGINAL SECTOR
                                        MOV     LSTRK,82$       ;STORE LAST TRACK.
                                        SWAB    82$             ;GET TRACK ADDRESS TO LO BYTE AND
                                        INC     82$             ;INCREMENT TO GET TOTL # OF TRACKS.

                                        BIC     #^CTADMSK,79$   ;SAVE TRACK ADDRESS BITS AND
                                        SWAB    79$             ;GET TRACK ADDRESS TO LO BYTE.
                                        BIC     #^CSADMSK,80$   ;SAVE SECTOR ADDRESS BITS
                                        ADD     (SP)+,80$

                                        BIT     #SSEI,RMOFO     ;WAS SSEI SET ?
                                        BEQ     61$             ;NO !!

                                        CMP     80$,#32.        ;SECTOR OVEFLOWED??
                                        BLO     63$             ;NO!!
                                        INC     79$             ;INCREMENT TRACK
                                        SUB     #32.,80$        ;ADJUST SECTOR

                                61$:    CMP     80$,#31.        ;SECTOR OVERFLOWED ?
                                        BLO     63$             ;NO !!
                                62$:    INC     79$             ;INCREMENT TRACK
                                        SUB     #31.,80$        ;ADJUST SECTOR
                                        BR      61$             ;TRY AGAIN

                                63$:    CMP     79$,82$         ;TRACK OVERFLOWED??
                                        BLO     64$             ;NO!!
                                        INC     78$             ;INCREMENT CYLINDER
                                        SUB     82$,79$         ;ADJUST TRACK
                                        BR      63$             ;TRY AGAIN
                                        NOP

                                ;REPORT ERROR IF "SSEI" IS SET AFTER TRACK SWITCHING OCCURED
                                ;THRU DATA TRANSFER
                                64$:    CLR     $GDDAT          ;SETUP EXPECTED SSEI = 0
                                        MOV     RMOFI,$BDDAT    ;GET RECIEVED DATA AND
                                        BIC     #^CSSEI,$BDDAT  ;SAVE SSEI BIT FOR COMPARE.

                                        BIT     #SSEI,RMOFO     ;WAS SSEI SET BEFORE DATA TRANSFER ?
                                        BEQ     66$             ;NO
```

```
045336  123737  001421  001345          CMPB    RMDA0+1,RMDAI+1 ;DID TRACK SWITCH OCCUR ?
045344  001411                          BEQ     65$             ;NO
045346  005737  001142                  TST     $BDDAT          ;IS SSEI STILL SET ?
045352  001430                          BEQ     68$             ;NO
045354  062716  000004                  ADD     #4,(SP)         ;POINT TO ERROR
045360  112776  000225  000000          MOVB    #225,@(SP)      ;DEFINE ERROR NUMBER
045366  000414                          BR      67$

                                ;REPORT ERROR IF "SSEI" IS INCORRECT
045370  012737  001000  001140  65$:    MOV     #SSEI,$GDDAT    ;SETUP EXPECTED SSEI = 1
045376  023737  001140  001142  66$:    CMP     $GDDAT,$BDDAT   ;IS SSEI CORRECT ?
045404  001413                          BEQ     68$             ;YES
045406  062716  000004                  ADD     #4,(SP)         ;POINT TO ERROR
045412  112776  000226  000000          MOVB    #226,@(SP)      ;DEFINE ERROR NUMBER
045420  162716  000002          67$:    SUB     #2,(SP)         ;ADJUST RETURN TO ERROR
045424  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
045426  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
045432  000240                          NOP

                                ;REPORT ERROR IF "LBT" IS NOT CORRECT
045434  005037  001140          68$:    CLR     $GDDAT          ;SET GOOD DATA FOR LBT = 0
045440  023727  046112  001060          CMP     78$,#560.       ;SHOULD LBT BE SET??
045446  101407                          BLOS    69$             ;NO!!
045450  032737  002000  001352          BIT     #IAE,RMER1I     ;WAS IAE SET ??
045456  001003                          BNE     69$             ;YES - LBT SHOULD NOT BE SET
045460  012737  002000  001140          MOV     #LBT,$GDDAT     ;SET GOOD DATA FOR LBT = 1
045466  013737  001350  001142  69$:    MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
045474  042737  175777  001142          BIC     #^CLBT,$BDDAT
045502  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS LBT CORRECT??
045510  001413                          BEQ     70$             ;YES!!
045512  062716  000004                  ADD     #4,(SP)
045516  112776  000017  000000          MOVB    #17,@(SP)       ;ERROR NUMBER
045524  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
045530  004736                          JSR     PC,@(SP)+       ;REPORT LBT IS WRONG
045532  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
045536  000240                          NOP

                                ;REPORT ERROR IF "AOE" IS INCORRECT
045540  005037  001140          70$:    CLR     $GDDAT          ;SET FOR AOE = 0
045544  032737  002000  001352          BIT     #IAE,RMER1I     ;WAS "IAE" DETECTED??
045552  001031                          BNE     72$             ;YES-"AOE" SHOULD BE ZERO
045554  023727  046112  001060          CMP     78$,#560.       ;SHOULD AOE BE SET??
045562  101425                          BLOS    72$             ;NO!!
045564  005737  046114                  TST     79$             ;MAYBE
045570  001012                          BNE     71$             ;YES
045572  005737  046116                  TST     80$
045576  001007                          BNE     71$             ;YES !!
045600  032737  000010  046120          BIT     #F2,81$ ;WAS THIS READ OR WRITE CHECK ??
045606  001413                          BEQ     72$             ;NO !!
045610  005737  001340                  TST     RMWCI           ;WAS ALL DATA TRANSFERRED ??
045614  001410                          BEQ     72$             ;YES !!
045616  012737  001000  001140  71$:    MOV     #AOE,$GDDAT     ;SET FOR AOE = 1
045624  005037  046114                  CLR     79$             ;CLEAR EXPECTED TRACK
045630  012737  000001  046116          MOV     #1,80$          ;EXPECT SECTOR = 1
045636  013737  001352  001142  72$:    MOV     RMER1I,$BDDAT   ;BAD DATA FOR TYPEOUT
045644  042737  176777  001142          BIC     #^CAOE,$BDDAT
045652  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS AOE CORRECTY??
```

```
    685 045660  001413                          BEQ     73$             ;YES!!
    686 045662  062716  000004                  ADD     #4,(SP)
    687 045666  112776  000020  000000          MOVB    #20,@(SP)       ;ERROR NUMBER
    688 045674  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    689 045700  004736                          JSR     PC,@(SP)+       ;REPORT AOE IS WRONG
    690 045702  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
    691 045706  000240                          NOP
    692
    693                             ;REPORT ERROR IF RMDA IS NOT CORRECT
    694 045710  032737  002000  001352  73$:    BIT     #IAE,RMER1I     ;WAS THERE AN IAE ERROR ??
    695 045716  001062                          BNE     75$             ;YES - DONT CHECK RMDA,RMDC
    696 045720  013737  046114  001140          MOV     79$,$GDDAT      ;SETUP EXPECTED DISK ADDRESS
    697 045726  000337  001140                  SWAB    $GDDAT
    698 045732  113737  046116  001140          MOVB    80$,$GDDAT
    699 045740  013737  001344  001142          MOV     RMDAI,$BDDAT    ;SETUP RECEIVED DISK ADDRESS
    700 045746  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
    701 045754  001413                          BEQ     74$             ;BRANCH IF EQUAL
    702 045756  062716  000004                  ADD     #4,(SP)
    703 045762  112776  000021  000000          MOVB    #21,@(SP)       ;ERROR NUMBER
    704 045770  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    705 045774  004736                          JSR     PC,@(SP)+       ;REPORT BAD DISK ADDRESS
    706 045776  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
    707 046002  000240                          NOP
    708
    709                             ;REPORT ERROR IF RMDC IS INCORRECT
    710 046004  013737  046112  001140  74$:    MOV     78$,$GDDAT      ;SETUP EXPECTED CYLINDER
    711 046012  042737  176000  001140          BIC     #^C1777,$GDDAT
    712 046020  013737  001372  001142          MOV     RMDCI,$BDDAT    ;SETUP RECEIVED CYLINDER
    713 046026  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE CYLINDERS
    714 046034  001413                          BEQ     75$             ;BRANCH IF EQUAL
    715 046036  062716  000004                  ADD     #4,(SP)
    716 046042  112776  000022  000000          MOVB    #22,@(SP)       ;ERROR NUMBER
    717 046050  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    718 046054  004736                          JSR     PC,@(SP)+       ;REPORT BAD CYLINDER
    719 046056  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
    720 046062  000240                          NOP
    721
    722 046064  062716  000004          75$:    ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
    723 046070  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
    724 046074  001403                          BEQ     76$
    725 046076  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR RETURN
    726 046102  000402                          BR      77$
    727 046104  162716  000004          76$:    SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    728 046110  000207                  77$:    RTS     PC
    729
    730 046112  000000                  78$:    .WORD   0               ;CYLINDER
    731 046114  000000                  79$:    .WORD   0               ;TRACK
    732 046116  000000                  80$:    .WORD   0               ;SECTOR
    733 046120  000000                  81$:    .WORD   0               ;FUNCTION CODE
    734 046122  000000                  82$:    .WORD   0               ;TOTAL # OF TRACKS = LAST TRACK +1
```

```
                                        .SBTTL  COMPOSITE ERROR CHECK SUBROUTINE
                                        ;THIS SUBROUTINE CHECKS THE STORED CONTENTS OF RMER1 AND
                                        ;RMER2 AFTER MASKING EACH REGISTER WORD WITH THE USER'S STATUS
                                        ;MASKS AND REPORTS AN ERROR IF ANY BITS ARE LEFT ON AFTER
                                        ;THE MASKS ARE APPLIED.

                                        ;CALL:
                                        ;(1)    JSR     PC,CMPERRSTS
                                        ;       .WORD                   MASK FOR ERROR REGISTER 1
                                        ;       .WORD                   MASK FOR ERROR REGISTER 2
                                        ;       BR      ???             RETURN HERE IF NO ERROR
                                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                        ;       ???                     RETURN HERE IF NO MORE ERRORS

                                        ;NOTE:  BITS TO BE MASKED SHOULD BE ONE; BITS TO BE TESTED SHOULD
                                        ;BE ZERO

046124                                  CMPERRSTS:

                                        ;MASK AND STORE THE CONTENTS OF RMER1  AND RMER2
046124  013737  001352  001176                 MOV     RMER1I,$TMP1    ;STORE RMER1 AT TEMP STORAGE
046132  047637  000000  001176                 BIC     @(SP),$TMP1     ;MASK   RMER1
046140  062716  000002                         ADD     #2,(SP)         ;MOVE SP TO NEXT MASK
046144  013737  001400  001200                 MOV     RMER2I,$TMP2    ;STORE RMER2 AT TEMP STORAGE
046152  047637  000000  001200                 BIC     @(SP),$TMP2     ;MASK RMER2


                                        ;CLEAR USER'S ERROR CALL
046160  062716  000006                         ADD     #6,(SP)         ;MOVE SP TO USER'S ERROR CALL
046164  105076  000000                         CLRB    @(SP)           ;CLEAR ERROR NUMBER
046170  162716  000004                         SUB     #4,(SP)         ;LEAVE SP AT NO ERROR RETURN

                                        ;SEE IF THERE WERE ANY ERRORS IN RMER1, I.E., $TMP1
046174  005737  001176                         TST     $TMP1           ;ANY ERRORS TO REPORT??
046200  001420                                 BEQ     1$              ;NO !!
046202  013737  001176  001142                 MOV     $TMP1,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
046210  005037  001140                         CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
046214  062716  000004                         ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
046220  112776  000066  000000                 MOVB    #66,@(SP)       ;CORRECTABLE DATA CHECK ERROR #
046226  162716  000002                         SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046232  004736                                 JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
046234  162716  000010                         SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
046240  000240                                 NOP
046242                                  1$:

                                        ;SEE IF THERE ARE ANY ERRORS TO REPORT IN RMER2 ($TMP2)
046242  005737  001200                         TST     $TMP2           ;ANY ERRORS IN RMER2?
046246  001420                                 BEQ     2$              ;NO!!

046250  013737  001200  001142                 MOV     $TMP2,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
046256  005037  001140                         CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
046262  062716  000004                         ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
046266  112776  000067  000000                 MOVB    #67,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
046274  162716  000002                         SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
```

COMPOSITE ERROR CHECK SUBROUTINE

```
     58 046300  004736                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     59 046302  162716  000010                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     60 046306  000240                        NOP
     61 046310                          2$:
     62
     63                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
     64 046310  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     65 046314  105716  000000                TSTB    @(SP)           ;WAS THERE AN ERROR CALLED??
     66 046320  001403                        BEQ     3$              ;NO!!
     67 046322  062716  000004                ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
     68 046326  000402                        BR      4$
     69 046330  162716  000004          3$:   SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
     70 046334  000207                  4$:   RTS     PC              ;RETURN TO USER
```

```
                        .SBTTL  DEVICE SELECT SUBROUTINE

                        ; THIS SUBROUTINE SELECTS THE DEVICE, GETTING THE DEVICE NUMBER FROM THE
                        ; TEST QUEUE.

                        ;CALL:
                        ;(1)    JSR     PC,DEVSEL
                        ;(2)    BR      ??              RETURN IF NO ERROR
                        ;(3)    NOP                     RETURN IF ERROR
                        ;(4)    ERROR                   ERROR DEFINED BY SUBROUTINE

046336                  DEVSEL:

                        ;CLEAR USER'S ERROR CALL
046336  062716  000004          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
046342  105076  000000          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF CALL
046346  162716  000004          SUB     #4,(SP)         ;MOVE SP BACK

                        ;SAVE USER'S INFORMATION AND SETUP REGISTERS
046352  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
046356  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
046362  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
046364  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
046366  012737  046506  000004  MOV     #2$,ERRVEC      ;SETUP FOR BUS TIMEOUT
046374  012737  000300  000006  MOV     #PR6,ERRVEC+2
046402  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
046406  013701  001466          MOV     TSTQUE,R1       ;R1 POINTS TO DEVICE NUMBER

                        ;SELECT DEVICE AND VERIFY THAT DEVICE IS AVAILABLE
046412  111160  000010          MOVB    (R1),RMCS2(R0)  ;WRITE UNIT SELECT BITS
046416  016037  000000  001176  MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
046424  016037  000010  001174  MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS

046432  032737  010000  001174  BIT     #NED,$TMP0      ;IS DEVICE NONEXISTENT ?
046440  001407                  BEQ     1$              ;NO!!
046442  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
046450  112776  000111  000010  MOVB    #111,@10(SP)    ;WRITE ERROR NUMBER
046456  000422                  BR      3$

046460  032737  004000  001176  1$:     BIT     #DVA,$TMP1      ;IS DEVICE AVAILABLE ?
046466  001021                  BNE     4$              ;YES!!
046470  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
046476  112776  000112  000010  MOVB    #112,@10(SP)    ;WRITE ERROR NUMBER
046504  000407                  BR      3$

                        ;HANDLE BUS TIMEOUT
046506  022626          2$:     CMP     (SP)+,(SP)+     ;ADJUST SP
046510  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
046516  112776  000113  000010  MOVB    #113,@10(SP)    ;WRITE BUS TIMEOUT ERROR NUMBER
046524  162766  000002  000010  3$:     SUB     #2,10(SP)       ;ADJUST RETURN TO 'NOP' PRECEDING
                                                        ;THE ERROR CALL

                        ;RESTORE USERS DATA AND RETURN TO ADDRESS ON STACK
046532                  4$:
046532  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
046534  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
046536  012637  000006          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
```

DEVICE SELECT SUBROUTINE

```
        046542  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     52 046546  000207                          RTS     PC              ;EXIT
```

```
                                        .SBTTL  SEEK STATUS CHECK SUBROUTINE
                                        ;
                                        ;THIS SUBROUTINE VERIFIES THE RESULTS OF SEEK TESTS USING STATUS
                                        ;STORED IN THE GET BUFFER AND TEST PARAMETERS STORED IN THE PUT BUFFER.
                                        ;
                                        ;
                                        ;THE SUBROUTINE RETURNS TO THE CALLING ROUTINE IF AN ERROR IS DETECTED
                                        ;AFTER HAVING LOADED THE APPROPRIATE ERROR NUMBER IN THE "ERROR" TRAP
                                        ;OF THE CALLING ROUTINE.  SEEK STATUS IS CHECKED AS FOLLOWS:
                                        ;
                                        ;CALL:
                                        ;(1)     JSR     PC,SEKSTS
                                        ;       BR      ???             RETURN HERE IF NO ERROR
                                        ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                        ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                        ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                        ;       ???                     RETURN HERE IF NO MORE ERRORS
                                        ;
046550                                  SEKSTS:
                                        ;
                                        ;CLEAR USERS' ERROR CALL
                                                NOP
046550  000240                                 ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
046552  062716  000004                         CLRB    @(SP)           ;CLEAR ERROR NUMBER
046556  105076  000000                         SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
046562  162716  000004                         CLR     20$             ;CLEAR ERROR FLAGS
046566  005037  050026
                                        ;
                                        ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN WRITING
                                        ;LOCAL REGISTERS, I.E., "PAR" = 1 AND "DPE" = 0
046572  032737  000010  001352                BIT     #PAR,RMER1I     ;WAS PARITY ERROR DETECTED??
046600  001424                                 BEQ     1$              ;NO!!
046602  032737  000010  001400                BIT     #DPE,RMER2I     ;WAS IT DUE TO CONTROL BUS??
046610  001020                                 BNE     1$              ;NOT SURE!!
                                        ;
                                        ;REPORT REGISTER PARITY ERROR VIA USER'S ERROR CALL
046612  005037  001140                         CLR     $GDDAT          ;EXPECTED STATUS
046616  013737  001352  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
046624  062716  000004                         ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
046630  112776  000050  000000                MOVB    #50,@(SP)       ;ERROR #50
046636  162716  000002                         SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046642  004736                                 JSR     PC,@(SP)+
046644  162716  000010                         SUB     #10,(SP)        ;RESTORE STACK
046650  000447                                 BR      3$              ;IAE SHOULD  BE ZERO
                                        ;
                                        ;DETERMINE THE VALUE OF "IAE" STATUS BASED ON TRACK, SECTOR, CYLINDER
                                        ;AND THE STATUS OF "SSEI" BIT DURING A SEEK OPERATION. ALSO, SET "SKI"
                                        ;IF CYLINDER ADDRESS IS TOO LARGE.
046652  012737  002000  001140        1$:     MOV     #IAE,$GDDAT     ;SETUP FOR IAE = 1
046660  052737  040000  050026                BIS     #SKI,20$        ;SETUP FOR SKI = 1
046666  023727  001446  001060                CMP     RMDC0,#560.     ;GREATER THAN LAST CYLINDER ?
046674  101035                                 BHI     3$              ;YES - CYLINDER IS INVALID
046676  042737  040000  050026                BIC     #SKI,20$        ;CLEAR SKI ERROR FLAG
                                        ;
046704  123737  001421  001335                CMPB    RMDA0+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
046712  101026                                 BHI     3$              ;YES - TRACK IS INVALID
                                        ;
046714  123727  001420  000035                CMPB    RMDA0,#29.      ;SECTOR > 29. ?
```

SEEK STATUS CHECK SUBROUTINE

```
                                                BLOS    2$              ;BR IF NO
                                                BIT     #FMT16,RMOFO    ;18 BIT FORMAT ?
                                                BEQ     3$              ;YES - SECTOR IS INVALID FOR 18 BIT MODE
                                                CMPB    RMDA0,#30.      ;SECTOR > 30. ?
                                                BLOS    2$              ;BR IF NO
                                                BIT     #SSEI,RMOFO     ;IS SSEI CLEAR ?
                                                BEQ     3$              ;YES - SECTOR IS INVALID FOR 16 BIT MODE
                                                CMPB    RMDA0,#31.      ;SECTOR > 31. ?
                                                BHI     3$              ;YES - SECTOR IS INVALID

                                        2$:     CLR     $GDDAT          ;"IAE" SHOULD = 0

                                        ;COMPARE EXPECTED AND RECIEVED "IAE" STATUS
                                        3$:     MOV     RMER1I,$BDDAT   ;IS IAE OK??
                                                BIC     #^CIAE,$BDDAT   ;SAVE IAE BIT FOR COMPARE
                                                CMP     $GDDAT,$BDDAT   ;CORRECT "IAE" STATUS ?
                                                BNE     4$              ;BR IF NO
                                                BIC     #SKI,20$        ;CLEAR SKI FLAG
                                                BR      5$              ;GO CHECK NEXT ERROR
                                        4$:
                                        ;REPORT INCORRECT "IAE" STATUS VIA USER'S ERROR CALL
                                                ADD     #4,(SP)
                                                MOVB    #51,@(SP)       ;ERROR 51
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT INCORRECT IAE
                                                SUB     #10,(SP)                ;RESTORE (SP)
                                                NOP
                                        5$:

                                        ;REPORT ANY IVC ERROR AS
                                        ;       IVC ERROR WITH VOLUME VALID ZERO
                                        ;       ERRONEOUS IVC ERROR, VOLUME VALID IS SET
                                                BIT     #IVC,RMER2I     ;IVC ERROR??
                                                BEQ     7$              ;NO!!
                                                CLR     $GDDAT          ;EXPECTED STATUS
                                                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                                MOVB    #60,@(SP)       ;ERROR 60 IF VV = 0
                                                BIT     #VV,RMDSI
                                                BEQ     6$
                                                MOVB    #61,@(SP)       ;ERROR 61 IF VV = 1
                                        6$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                                SUB     #10,(SP)        ;RESTORE SP
                                                NOP

                                        7$:     MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                                BIC     #^CSKI,$BDDAT   ;CLEAR DONT CARES
                                                MOV     20$,$GDDAT      ;GET EXPECTED SKI STATUS
                                                BIC     #^CSKI,$GDDAT   ;CLEAR DONT CARES
                                                BEQ     8$              ;BRANCH IF 0 EXPECTED

                                        ;REPORT ERROR IF SKI IS NOT SET (IAE WAS NOT DETECTED)
                                                BIT     #SKI,$BDDAT     ;WAS SKI DETECTED ??
                                                BNE     9$              ;YES !!
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #267,@(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
                                        BR      10$             ;GO TO NEXT ERROR CHECK
                                8$:

                                ;REPORT ERROR IF SKI IS SET
                                        BIT     #SKI,$BDDAT     ;IS SKI SET ??
                                        BEQ     9$              ;NO - SKI IS OK
                                        ADD     #4,(SP)         ;MOVE (SP) TO ERROR
                                        MOVB    #54,@(SP)       ;LOAD ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT SEEK ERROR
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP

                                ;REPORT ANY DEVICE CHECK
                                9$:     BIT     #DVC,RMER2I     ;WAS THERE DVC DURING SEEK??
                                        BEQ     10$             ;NO!!
                                        CLR     $GDDAT          ;EXPECTED STATUS
                                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)
                                        MOVB    #55,@(SP)       ;ERROR  #55
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                        SUB     #10,(SP)        ;RESTORE SP
                                        NOP

                                ;REPORT ANY "OPI" ERROR AS OPI WITH MOL  = 0, OR  OPI
                                ;BECAUSE ON CYLINDER LATCH DIDN'T RESET
                                10$:    BIT     #OPI,RMER1I     ;"OPI" ERROR??
                                        BEQ     12$             ;NO!!
                                        CLR     $GDDAT          ;EXPECTED STATUS
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE (SP) TO ERROR
                                        MOVB    #52,@(SP)       ;LOAD ERROR NUMBER
                                        BIT     #MOL,RMDSI      ;WAS MEDIUM ON LINE??
                                        BEQ     11$             ;NO!!
                                        MOVB    #53,@(SP)       ;YES - CHANGE ERROR NUMBER
                                11$:    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT "OPI" ERROR
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP

                                ;SEE IF "PIP" = 0, AND "ATA", "MOL" AND "VV" = 1
                                12$:    MOV     RMDSI,-(SP)
                                        BIC     #^C<ATA!PIP!MOL!VV>,(SP)
                                        CMP     #ATA!MOL!VV,(SP)+
                                        BNE     13$             ;ERROR IN RMDS
                                        JMP     17$             ;RMDS IS OK

                                ;REPORT ERROR IF MOL = 0 AND OPI = 0
                                13$:    BIT     #MOL,RMDSI      ;IS MOL RESET??
                                        BNE     14$             ;NO - MOL IS SET
                                        BIT     #OPI,RMER1I     ;WAS OPI SET
                                        BNE     14$             ;YES - DONT REPORT ERROR
                                        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
```

```
                                        BIS     #MOL,$GDDAT
                                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                        ADD     #4,(SP)
                                        MOVB    #62,@(SP)
                                        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
                        14$:    BIT     #PIP,RMDSI      ;IS "PIP" STILL SET??
                                BEQ     15$             ;NO!!
                                BIT     #SKI,RMER2I     ;WAS "SKI"SET??
                                BNE     15$             ;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
                        15$:    BIT     #ATA,RMDSI      ;WAS "ATA" SET ??
                                BNE     16$             ;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
                        16$:    BIT     #VV,RMDSI       ;IS VV = 0 ??
                                BNE     17$             ;NO!!
                                BIT     #IVC,RMER2I     ;IS IVC ALSO 0 ??
                                BNE     17$             ;NO - IVC IS SET
                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                BIS     #VV,$GDDAT
                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                ADD     #4,(SP)
                                MOVB    #64,@(SP)       ;ERROR #64
                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                JSR     PC,@(SP)+
                                SUB     #10,(SP)
                                NOP
                        17$:

                                ;MODIFY THE RETURN ADDRESS IF AN ERROR WAS DETECTED
                                NOP
                                ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
```

SEEK STATUS CHECK SUBROUTINE

```
    229 050004  105776  000000              TSTB    @(SP)           ;WAS ERROR CALLED??
    230 050010  001403                      BEQ     18$             ;NO!!
    231 050012  062716  000004              ADD     #4,(SP)         ;MOVE TO ERROR RETURN
    232 050016  000402                      BR      19$
    233
    234 050020  162716  000004      18$:    SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    235 050024  000207              19$:    RTS     PC              ;RETURN
    236
    237 050026  000000              20$:    .WORD   0               ;ERROR FLAGS
```

CONTROLLER CLEAR SUBROUTINE

```
                                        .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
                                        ;       ???
                                        ;
050030                                  CNTCLR:
050030  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
050032  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
050034  013746  000004                       MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
050040  013746  000006                       MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
050044  012737  050104  000004               MOV     #1$,ERRVEC      ;SETUP FOR BUS TIMEOUT
050052  012737  000300  000006               MOV     #PR6,ERRVEC+2
050060  013700  001276                       MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
050064  012760  000040  000010               MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
050072  013701  001466                       MOV     TSTQUE,R1       ;GET DEVICE UNDER TEST
050076  111160  000010                       MOVB    (R1),RMCS2(R0)  ;SELECT DEVICE
050102  000412                               BR      2$
                                        ;
050104  022626                        1$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
050106  062766  000004  000010               ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
050114  112776  000007  000010               MOVB    #7,@10(SP)      ;WRITE THE ERROR NUMBER
050122  162766  000002  000010               SUB     #2,10(SP)       ;ADJUST SP TO RETURN TO ERROR
050130                                  2$:
050130  012637  000006                       MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
050134  012637  000004                       MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
050140  012601                               MOV     (SP)+,R1        ;;POP STACK INTO R1
050142  012600                               MOV     (SP)+,R0        ;;POP STACK INTO R0
050144  000207                               RTS     PC
```

CONTROLLER CLEAR STATUS CHECK SUBROUTINE

```
                                        .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

050146                                  CLRSTS:

                                        ;CLEAR USER'S ERROR CALL
050146  062716  000004                          ADD     #4,(SP)                 ;MOVE SP TO ERROR
050152  105076  000000                          CLRB    @(SP)                   ;CLEAR ERROR NUMBER
050156  162716  000004                          SUB     #4,(SP)                 ;MOVE SP BACK TO NO ERROR
                                        ;REPORT ERROR IF RMCS1 NOT INITIALIZED
050162  013737  001336  001142          1$:     MOV     RMCS1I,$BDDAT           ;VERIFY RMCS1
050170  042737  100000  001142                  BIC     #SC,$BDDAT              ;IGNORE SPECIAL CONDITION
050176  012737  004200  001140                  MOV     #DVA!RDY,$GDDAT         ;EXPECT DVA & RDY
050204  023737  001140  001142                  CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED, RECEIVED
050212  001413                                  BEQ     2$                      ;BRANCH IF EQUAL
050214  062716  000004                          ADD     #4,(SP)                 ;MOVE SP TO USER'S ERROR CALL
050220  112776  000126  000000                  MOVB    #126,@(SP)              ;WITE ERROR NUMBER IN CALL
050226  162716  000002                          SUB     #2,(SP)                 ;MOVE SP TO RETURN FOR ERROR
050232  004736                                  JSR     PC,@(SP)+               ;REPORT ERROR VIA USER
050234  162716  000010                          SUB     #10,(SP)                ;MOVE SP BACK TO NO ERROR
050240  000240                                  NOP
                                        ;REPORT ERROR IF RMBA NOT RESET
050242  005037  001140          2$:     CLR     $GDDAT                  ;VERIFY RMBA IS ZERO
050246  013737  001342  001142                  MOV     RMBAI,$BDDAT
050254  001413                                  BEQ     3$                      ;BRANCH IF ZERO
050256  062716  000004                          ADD     #4,(SP)                 ;MOVE SP TO USER'S ERROR CALL
050262  112776  000127  000000                  MOVB    #127,@(SP)              ;WITE ERROR NUMBER IN CALL
050270  162716  000002                          SUB     #2,(SP)                 ;MOVE SP TO RETURN FOR ERROR
050274  004736                                  JSR     PC,@(SP)+               ;REPORT ERROR VIA USER
050276  162716  000010                          SUB     #10,(SP)                ;MOVE SP BACK TO NO ERROR
050302  000240                                  NOP
                                        ;REPORT ERROR IF RMCS2 NOT INITIALIZED
050304  013737  001346  001142          3$:     MOV     RMCS2I,$BDDAT           ;VERIFY RMCS2
050312  010146                                  MOV     R1,-(SP)                ;;PUSH R1 ON STACK
050314  005046                                  CLR     -(SP)                   ;EXPECT IR & UNIT NUMBER
050316  013701  001466                          MOV     TSTQUE,R1               ;R1 = ADDRESS OF TEST QUE
050322  111116                                  MOVB    (R1),(SP)
050324  052716  000100                          BIS     #IR,(SP)
050330  012637  001140                          MOV     (SP)+,$GDDAT
```

```
 58 050334  012601                              MOV     (SP)+,R1        ;;POP STACK INTO R1
 59 050336  023737  001140  001142              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 60 050344  001413                              BEQ     4$              ;BRANCH IF EQUAL
 61 050346  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 62 050352  112776  000130  000000              MOVB    #130,@(SP)      ;WITE ERROR NUMBER IN CALL
 63 050360  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 64 050364  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 65 050366  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 66 050372  000240                              NOP
 67                                     ;REPORT ERROR IF RMER1 NOT RESET-IGNORE UNS
 68 050374  005037  001140              4$:     CLR     $GDDAT          ;VERIFY RMER1
 69 050400  013737  001352  001142              MOV     RMER1I,$BDDAT
 70 050406  042737  040000  001142              BIC     #UNS,$BDDAT     ;IGNORE UNSAFE
 71 050414  001413                              BEQ     5$              ;BRANCH IF ZERO
 72 050416  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 73 050422  112776  000131  000000              MOVB    #131,@(SP)      ;WITE ERROR NUMBER IN CALL
 74 050430  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 75 050434  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 76 050436  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 77 050442  000240                              NOP
 78                                     ;REPORT ERROR IF RMMR1 NOT INITIALIZED-IGNORE WC,LS,LST
 79 050444  013737  001362  001142      5$:     MOV     RMMR1I,$BDDAT   ;VERIFY RMMR
 80 050452  042737  000046  001142              BIC     #WC!LS!LST,$BDDAT       ;IGNORE WORD CLOCK, SCT, TRK
 81 050460  012737  000010  001140              MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA BIT
 82 050466  023737  001140  001142              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
 83 050474  001413                              BEQ     6$              ;BRANCH IF 0
 84 050476  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 85 050502  112776  000133  000000              MOVB    #133,@(SP)      ;WITE ERROR NUMBER IN CALL
 86 050510  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 87 050514  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 88 050516  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 89 050522  000240                              NOP
 90                                     ;REPORT AN ERROR IF RMEC2 IS NOT RESET
 91 050524  005037  001140              6$:     CLR     $GDDAT          ;EXPECT ZEROS
 92 050530  013737  001404  001142              MOV     RMEC2I,$BDDAT   ;VERIFY RMEC2=0
 93 050536  001413                              BEQ     7$
 94 050540  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 95 050544  112776  000135  000000              MOVB    #135,@(SP)      ;WITE ERROR NUMBER IN CALL
 96 050552  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 97 050556  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 98 050560  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 99 050564  000240                              NOP
100                                     ;REPORT ERROR IF RMMR2 NOT INITIALIZED-IGNORE RQA,RQB
101 050566  013737  001376  001142      7$:     MOV     RMMR2I,$BDDAT   ;VERIFY RMMR2
102 050574  042737  140000  001142              BIC     #RQA:RQB,$BDDAT
103 050602  012737  011777  001140              MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
104 050610  023737  001140  001142              CMP     $GDDAT,$BDDAT
105 050616  001413                              BEQ     8$
106 050620  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
107 050624  112776  000136  000000              MOVB    #136,@(SP)      ;WITE ERROR NUMBER IN CALL
108 050632  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
109 050636  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
110 050640  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
111 050644  000240                              NOP
112                                     ;REPORT ERROR IF RMER2 NOT RESET-IGNORE SKI,DVC
113 050646  005037  001140              8$:     CLR     $GDDAT          ;EXPECT ALL ZEROS
114 050652  013737  001400  001142              MOV     RMER2I,$BDDAT   ;VERIFY RMER2
```

CONTROLLER CLEAR STATUS CHECK SUBROUTINE

```
    115 050660  042737  040200  001142          BIC     #SKI!DVC,$BDDAT ;IGNORE DEVICE ERRORS
    116 050666  001413                          BEQ     9$              ;BRANCH IF OTHER BITS 0
    117 050670  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    118 050674  112776  000174  000000          MOVB    #174,@(SP)      ;WITE ERROR NUMBER IN CALL
    119 050702  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    120 050706  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    121 050710  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    122 050714  000240                          NOP
    123                             ;REPORT ERROR IF RMDS NOT INITIALIZED
    124 050716  013737  001350  001142  9$:     MOV     RMDSI,$BDDAT    ;TEST DRIVE STATUS REGISTER
    125 050724  042737  177177  001142          BIC     #^C<DRY!DPR>,$BDDAT
    126 050732  012737  000600  001140          MOV     #DPR!DRY,$GDDAT ;EXPECTED DRIVE STATUS
    127 050740  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
    128 050746  001413                          BEQ     10$             ;BRANCH IF EQUAL
    129 050750  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    130 050754  112776  000134  000000          MOVB    #134,@(SP)      ;WRITE ERROR NUMBER
    131 050762  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    132 050766  004736                          JSR     PC,@(SP)+       ;REPORT ERROR TO USER
    133 050770  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    134 050774  000240                          NOP
    135 050776  062716  000004          10$:    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    136 051002  105776  000000                  TSTB    @(SP)           ;WAS AN ERROE DETECTED??
    137 051006  001403                          BEQ     11$             ;NO!!
    138 051010  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
    139 051014  000402                          BR      12$
    140 051016  162716  000004          11$:    SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
    141 051022  000240                  12$:    NOP
    142 051024  000207                          RTS     PC
```

```
                                .SBTTL  PACK ACKNOWLEDGE STATUS CHECK

                                ;THIS SUBROUTINE CHECKS THE RESULTS OF A PACK ACKNOWLEDGE
                                ;COMMAND USING THE STATUS STORED IN THE GET BUFFER.  ERRORS ARE
                                ;REPORTED TO THE USER VIA THE USER'S ERROR CALL.

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

051026                          ACKSTS:

                                ;CLEAR USER'S ERROR CALL
051026  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
051032  105076  000000                  CLRB    @(SP)           ;CLEAR LOW ORDER BYTE
051036  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK

                                ;REPORT AN ERROR  IF "VV" IS 0
051042  032737  000100  001350          BIT     #VV,RMDSI       ;IS VOLUME VALID SET??
051050  001024                          BNE     1$              ;YES!!
051052  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
051060  052737  000100  001140          BIS     #VV,$GDDAT
051066  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
051074  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
051100  112776  000155  000000          MOVB    #155,@(SP)      ;WRITE NUMBER IN ERROR CALL
051106  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051112  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR
051114  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
051120  000240                          NOP
051122                          1$:

                                ;REPORT AN ERROR IF "MOL" IS 0
051122  032737  010000  001350          BIT     #MOL,RMDSI      ;IS MOL SET??
051130  001024                          BNE     2$              ;YES!!
051132  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
051140  052737  010000  001140          BIS     #MOL,$GDDAT
051146  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
051154  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
051160  112776  000041  000000          MOVB    #41,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
051166  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051172  004736                          JSR     PC,@(SP)+       ;REPORT TH ERROR
051174  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO BRANCH
051200  000240                          NOP
051202                          2$:

                                ;SEE IF "UNS","OPI","RMR","ILR", OR "ILF" IS SET
051202  032737  060007  001352          BIT     #UNS!OPI!RMR!ILR!ILF,RMER1I
051210  001570                          BEQ     7$

                                ;REPORT AN ERROR IF "UNS" IS SET
051212  032737  040000  001352          BIT     #UNS,RMER1I     ;WAS UNS SET??
051220  001424                          BEQ     3$              ;NO!!
051222  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

PACK ACKNOWLEDGE STATUS CHECK

```
     58 051230  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
     59 051236  042737  040000  001140          BIC     #UNS,$GDDAT
     60 051244  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
     61 051250  112776  000042  000000          MOVB    #42,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
     62 051256  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     63 051262  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
     64 051264  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     65 051270  000240                          NOP
     66 051272                          3$:
     67
     68                                 ;REPORT ANY OPI ERROR
     69 051272  032737  020000  001352          BIT     #OPI,RMER1I     ;WAS OPI SET ??
     70 051300  001424                          BEQ     4$              ;NO!!
     71 051302  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
     72 051310  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
     73 051316  042737  020000  001140          BIC     #OPI,$GDDAT
     74 051324  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
     75 051330  112776  000043  000000          MOVB    #43,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
     76 051336  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     77 051342  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
     78 051344  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     79 051350  000240                          NOP
     80 051352                          4$:
     81
     82                                 ;REPORT ANY RMR ERROR
     83 051352  032737  000004  001352          BIT     #RMR,RMER1I     ;WAS RMR SET??
     84 051360  001424                          BEQ     5$              ;NO!!
     85 051362  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
     86 051370  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
     87 051376  042737  000004  001140          BIC     #RMR,$GDDAT
     88 051404  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
     89 051410  112776  000044  000000          MOVB    #44,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
     90 051416  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     91 051422  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
     92 051424  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     93 051430  000240                          NOP
     94 051432                          5$:
     95
     96                                 ;REPORT ANY     ILR ERROR
     97 051432  032737  000002  001352          BIT     #ILR,RMER1I     ;WAS ILR SET??
     98 051440  001424                          BEQ     6$              ;NO!!
     99 051442  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    100 051450  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
    101 051456  042737  000002  001140          BIC     #ILR,$GDDAT
    102 051464  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    103 051470  112776  000045  000000          MOVB    #45,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
    104 051476  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    105 051502  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
    106 051504  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
    107 051510  000240                          NOP
    108 051512                          6$:
    109
    110                                 ;REPORT ANY     ILF ERROR
    111 051512  032737  000001  001352          BIT     #ILF,RMER1I     ;WAS ILF SET??
    112 051520  001424                          BEQ     7$              ;NO!!
    113 051522  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    114 051530  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
```

```
    115 051536  042737  000001  001140          BIC     #ILF,$GDDAT
    116 051544  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    117 051550  112776  000046  000000          MOVB    #46,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
    118 051556  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    119 051562  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
    120 051564  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
    121 051570  000240                          NOP
    122 051572                          7$:
    123
    124                                 ;AUGMENT RETURN ADDRESS IF ERROR WAS FOUND
    125 051572  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    126 051576  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
    127 051602  001403                          BEQ     8$              ;NO!!
    128 051604  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
    129 051610  000402                          BR      9$
    130 051612  162716  000004          8$:     SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
    131 051616  000240                  9$:     NOP
    132 051620  000207                          RTS     PC
```

```
                                .SBTTL  RECALIBRATE STATUS CHECK SUBROUTINE
                                ;
                                ;THIS SUBROUTINE CHECKS THE RESULTS OF A RECALIBRATE OPERATION
                                ;USING THE STATUS STORED IN THE GET BUFFER.
                                ;
                                ;CALL:
                                ;
                                ;(1)     JSR     PC,RCLSTS       ;CALL SUBROUTINE
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS
                                ;
051622                          RCLSTS:
                                ;
                                ;CLEAR USER'S ERROR NUMBER
051622  062716  000004                  ADD     #4,(SP)
051626  105076  000000                  CLRB    @(SP)           ;CLEAR USER'S ERROR CALL
051632  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO BRANCH
                                ;
                                ;SEE IF "PAR" OR "ILF" OR "OPI" OR "IAE" IS SET
051636  032737  022011  001352          BIT     #OPI!PAR!ILF!IAE,RMER1I
051644  001553                  BEQ     5$              ;NONE ARE SET - GO TO NEXT CHECK
                                ;
                                ;REPORT ANY MASSBUS CONTROL BUS PARITY ERROR, I.E.,
                                ;"PAR" = 1 AND "DPE" = 0
051646  032737  000010  001352          BIT     #PAR,RMER1I     ;WAS "PAR" SET??
051654  001430                  BEQ     1$              ;NO!!
051656  032737  000010  001400          BIT     #DPE,RMER2I     ;WAS "DPE" SET??
051664  001024                  BNE     1$              ;YES - NOT A REGISTER ERROR
051666  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
051674  042737  000010  001140          BIC     #PAR,$GDDAT
051702  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
051710  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
051714  112776  000050  000000          MOVB    #50,@(SP)       ;WRITE ERROR NUMBER IN CALL
051722  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051726  004736                  JSR     PC,@(SP)+       ;GO REPORT ERROR
051730  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
051734  000240                  NOP
051736                          1$:
                                ;
                                ;REPORT ANY "ILF" ERROR
051736  032737  000001  001352          BIT     #ILF,RMER1I     ;WAS "ILF" SET??
051744  001424                  BEQ     2$              ;NO!!
051746  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
051754  042737  000001  001140          BIC     #ILF,$GDDAT
051762  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
051770  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
051774  112776  000071  000000          MOVB    #71,@(SP)       ;WRITE ERROR NUMBER IN CALL
052002  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052006  004736                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052010  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
052014  000240                  NOP
052016                          2$:
                                ;
```

```
                                        ;REPORT ANY "OPI" ERROR AS
                                        ;       . OPI DUE TO "MOL" = 0
                                        ;       . OPI BECAUSE ON CYLINDER LATCH DIDN'T RESET
052016  032737  020000  001352                BIT     #OPI,RMER1I     ;WAS OPI SET??
052024  001433                              BEQ     4$              ;NO!!
052026  013737  001352  001140              MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
052034  042737  020000  001140              BIC     #OPI,$GDDAT
052042  013737  001352  001142              MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
052050  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052054  112776  000072  000000              MOVB    #72,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
052062  032737  010000  001350              BIT     #MOL,RMDSI      ;WAS "MOL" = 0??
052070  001403                              BEQ     3$              ;YES!!
052072  112776  000073  000000              MOVB    #73,@(SP)       ;NO - CHANGE ERROR NUMBER
052100  162716  000002          3$:         SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052104  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052106  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
052112  000240                              NOP
052114                          4$:

                                        ;REPORT AN ERROR IF "IAE" IS SET
052114  032737  002000  001352              BIT     #IAE,RMER1I     ;IS "IAE" SET??
052122  001424                              BEQ     5$              ;NO!!
052124  013737  001352  001140              MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
052132  042737  002000  001140              BIC     #IAE,$GDDAT
052140  013737  001352  001142              MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
052146  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
052152  112776  000070  000000              MOVB    #70,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
052160  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052164  004736                              JSR     PC,@(SP)+       ;REPORT ERROR
052166  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
052172  000240                              NOP
052174                          5$:

                                        ;SEE IF "SKI" OR "IVC" OR "DVC" IS SET
052174  032737  050200  001400              BIT     #SKI!IVC!DVC,RMER2I
052202  001517                              BEQ     9$              ;NONE OF THE BITS ARE SET


                                        ;REPORT ANY "IVC" ERROR AS
                                        ;       .IVC WITH VV = 0
                                        ;       .ERRONEOUS IVC ERROR
052204  032737  010000  001400              BIT     #IVC,RMER2I     ;WAS IVC SET??
052212  001433                              BEQ     7$              ;NO!!
052214  013737  001400  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
052222  042737  010000  001140              BIC     #IVC,$GDDAT
052230  013737  001400  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
052236  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052242  112776  000074  000000              MOVB    #74,@(SP)       ;WRITE ERROR NUMBER IN CALL
052250  032737  000100  001350              BIT     #VV,RMDSI       ;WAS VV = 0??
052256  001403                              BEQ     6$              ;YES!!
052260  112776  000075  000000              MOVB    #75,@(SP)       ;NO - CHANGE ERROR NUMBER
052266  162716  000002          6$:         SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052272  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052274  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
052300  000240                              NOP
052302                          7$:

```

```
                                        ;REPORT ANY "SKI" ERROR
052302  032737  040000  001400                BIT     #SKI,RMER2I     ;WAS SKI SET??
052310  001424                                BEQ     8$              ;NO!!
052312  013737  001400  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
052320  042737  040000  001140                BIC     #SKI,$GDDAT
052326  013737  001400  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
052334  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052340  112776  000076  000000                MOVB    #76,@(SP)       ;WRITE ERROR NUMBER
052346  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052352  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052354  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO BRANCH
052360  000240                                NOP
052362                                  8$:

                                        ;REPORT ANY  "DVC" ERROR
052362  032737  000200  001400                BIT     #DVC,RMER2I     ;WAS "DVC" SET??
052370  001424                                BEQ     9$              ;NO!!
052372  013737  001400  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
052400  042737  000200  001140                BIC     #DVC,$GDDAT
052406  013737  001400  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
052414  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052420  112776  000077  000000                MOVB    #77,@(SP)       ;WRITE ERROR NUMBER
052426  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052432  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052434  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
052440  000240                                NOP
052442                                  9$:

                                        ;SEE IF "PIP" AND "OM" ARE 0, AND "ATA","MOL" AND "VV" ARE 1
052442  013746  001350                        MOV     RMDSI,-(SP)     ;PUT RMDS ON STACK
052446  042716  047676                        BIC     #^C<PIP!MOL!VV!OM!ATA>,(SP)
052452  022726  110100                        CMP     #ATA!MOL!VV,(SP)+
052456  001002                                BNE     10$
052460  000137  053074                        JMP     15$
052464                                  10$:

                                        ;REPORT AN ERROR IF MOL = 0 AND OPI = 0, I.E., MEDIUM WENT OFF
                                        ;LINE AFTER RECALIBRATE WAS INITIATED
052464  032737  010000  001350                BIT     #MOL,RMDSI      ;DID MOL DROP??
052472  001030                                BNE     11$             ;NO!!
052474  032737  020000  001352                BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
052502  001024                                BNE     11$             ;YES - DON'T REPORT MOL=0
052504  013737  001350  001140                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
052512  052737  010000  001140                BIS     #MOL,$GDDAT
052520  013737  001350  001142                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
052526  062716  000004                        ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
052532  112776  000100  000000                MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
052540  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052544  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
052546  162716  000010                        SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
052552  000240                                NOP
052554                                  11$:

                                        ;REPORT AN ERROR IF "VV" = 0 AND "IVC" = 0
052554  032737  000100  001350                BIT     #VV,RMDSI       ;DID "VV" DROP??
052562  001030                                BNE     12$             ;NO!!
052564  032737  010000  001400                BIT     #IVC,RMER2I     ;WAS THERE A IVC ERROR??
```

```
                                                BNE     12$             ;YES - DONT REPORT VV=0
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                BIS     #VV,$GDDAT
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #101,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+
                                                SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
                                                NOP
                                        12$:

                                        ;REPORT AN ERROR IF ATA IS NOT SET
                                                BIT     #ATA,RMDSI      ;WAS ATA SET DURING RECALIBRATE??
                                                BNE     13$             ;YES!!
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIS     #ATA,$GDDAT
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #102,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)
                                                JSR     PC,@(SP)+
                                                SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
                                                NOP

                                        13$:

                                        ;REPORT AN ERROR IF "OM" IS NOT ZERO BECAUSE RECALIBRATE SHOULD
                                        ;ALWAYS CLEAR OFFSET MODE
                                                BIT     #OM,RMDSI       ;WAS "OM" RESET??
                                                BEQ     14$             ;YES!!
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIC     #OM,$GDDAT
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #103,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO USER'S BR   CH
                                                NOP
                                        14$:

                                        ;REPORT AN ERROR IF "PIP" IS STIL ON, I.E., DRIVE NOT ON
                                        ;CYLINDER
                                                BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
                                                BEQ     15$             ;NO!!
                                                BIT     #SKI,RMER2I     ;WAS "SKI" DETECTED??
                                                BNE     15$             ;YES-DONT REPORT "PIP"
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIC     #PIP,$GDDAT
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #104,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+
                                                SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
                                                NOP
```

```
                                        15$:

                                        ;SEE IF "ILR" OR "RMR" OR "UNS" IS SET
053074  032737  040006  001352                BIT     #ILR!RMR!UNS,RMER1I
053102  001514                                BEQ     18$

                                        ;REPORT AN ERROR IF "ILR" IS SET
053104  032737  000002  001352                BIT     #ILR,RMER1I     ;WAS ILR SET DURING RECALIBRATE??
053112  001424                                BEQ     16$             ;NO!!
053114  013737  001352  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053122  042737  000002  001140                BIC     #ILR,$GDDAT
053130  013737  001352  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053136  062716  000004                        ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
053142  112776  000105  000000                MOVB    #105,@(SP)      ;WRITE ERROR NUMBER IN CALL
053150  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053154  004736                                JSR     PC,@(SP)+
053156  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
053162  000240                                NOP
053164                                  16$:

                                        ;REPORT AN ERROR IF "RMR" IS SET
053164  032737  000004  001352                BIT     #RMR,RMER1I     ;WAS RMR SET??
053172  001424                                BEQ     17$             ;NO!!
053174  013737  001352  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053202  042737  000004  001140                BIC     #RMR,$GDDAT
053210  013737  001352  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053216  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053222  112776  000106  000000                MOVB    #106,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
053230  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053234  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053236  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
053242  000240                                NOP
053244                                  17$:

                                        ;REPORT AN ERROR IF "UNS" IS SET AND "DVC" IS 0
053244  032737  040000  001352                BIT     #UNS,RMER1I     ;WAS UNSAFE ON??
053252  001430                                BEQ     18$             ;NO!!
053254  032737  000200  001400                BIT     #DVC,RMER2I     ;WAS THERE A DEVICE CHECK??
053262  001024                                BNE     18$             ;YES - DON'T REPORT UNSAFE
053264  013737  001352  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
053272  042737  040000  001140                BIC     #UNS,$GDDAT
053300  013737  001352  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053306  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053312  112776  000107  000000                MOVB    #107,@(SP)      ;WRITE ERROR NUMBER
053320  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053324  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053326  162716  000010                        SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
053332  000240                                NOP
053334                                  18$:

                                        ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
053334  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053340  105776  000000                        TSTB    @(SP)           ;WAS AN ERROR REPORTED??
053344  001403                                BEQ     19$             ;NO!!
053346  062716  000004                        ADD     #4,(SP)         ;YES - AUGMENT SP RETURN
053352  000402                                BR      20$
053354  162716  000004                19$:    SUB     #4,(SP) ;NO ERROR - RETURN SP TO BRANCH
```

```
    286 053360  000240                20$:    NOP
    287 053362  000207                        RTS     PC              ;STATUS CECK IS COMPLETE
```

```
                                        .SBTTL  DRIVE CLEAR STATUS CHECK SUBROUTINE
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS

053364                          DRVSTS:

                                ;CLEAR USER'S ERROR CALL
053364  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053370  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR CALL
053374  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO USER'S BRANCH
                                ;REPORT ERROR IF RMCS1 NOT INITIALIZED
053400  013737  001336  001142  1$:     MOV     RMCS1I,$BDDAT   ;CHECK RMCS1
053406  042737  173700  001142          BIC     #^C<DVA!FNCMSK>,$BDDAT  ;CLEAR DONT CARES
053414  012737  004010  001140          MOV     #DVA!DRVCLR,$GDDAT      ;EXPECT DVA
053422  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
053430  001443                          BEQ     3$              ;BRANCH IF EQUAL
053432  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053436  112776  000141  000000          MOVB    #141,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
053444  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053450  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
053452  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
053456  000240                          NOP
                                ;REPORT ERROR IF RMDS NOT INITIALIZED
053460  013737  001350  001142  2$:     MOV     RMDSI,$BDDAT    ;CHECK RMDS
053466  042737  021101  001142          BIC     #PGM!OM!VV!PIP,$BDDAT   ;CLEAR DONT CARES
053474  012737  010600  001140          MOV     #MOL!DPR!DRY,$GDDAT     ;EXPECT DRY & DPR
053502  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
053510  001413                          BEQ     3$              ;BRANCH IF EQUAL
053512  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053516  112776  000142  000000          MOVB    #142,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
053524  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053530  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
053532  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
053536  000240                          NOP
                                ;REPORT ERROR IF RMER1 NOT INITIALIZED
053540  005037  001140          3$:     CLR     $GDDAT          ;EXPECT 0'S
053544  013737  001352  001142          MOV     RMER1I,$BDDAT   ;CHECK RMER1
053552  001413                          BEQ     4$              ;BRANCH IF EQUAL
053554  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
053560  112776  000143  000000          MOVB    #143,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
053566  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053572  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
053574  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
053600  000240                          NOP
                                ;REPORT ERROR IF ATA NOT INITIALIZED
053602  013737  001354  001142  4$:     MOV     RMASI,$BDDAT    ;CHECK ATTENTION BIT
053610  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
053612  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
053614  013701  001466                  MOV     TSTQUE,R1
053620  116102  000001                  MOVB    1(R1),R2
053624  042702  177400                  BIC     #^CATNMSK,R2
053630  005102                          COM     R2
053632  040237  001142                  BIC     R2,$BDDAT
```

DRIVE CLEAR STATUS CHECK SUBROUTINE

```
                                                MOV     (SP)+,R2        ;;POP STACK INTO R2
                                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                                TST     $BDDAT          ;IS ATTENTION CLEARED??
                                                BEQ     5$              ;BRANCH IF ATTENTION CLEARED
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #144,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        ;REPORT ERROR IF RMMR1 NOT INITIALIZED
                                        5$:     MOV     RMMR1I,$BDDAT   ;CHECK RMMR
                                                BIC     #WC!LS!LST,$BDDAT       ;CLEAR DONT CARES
                                                MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA ON
                                                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                                BEQ     6$              ;BRANCH IF ZERO
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #145,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        ;REPORT ERROR IF RMMR2 NOT INITIALIZED
                                        6$:     MOV     RMMR2I,$BDDAT   ;CHECK RMMR2
                                                BIC     #RQA!RQB,$BDDAT ;CLEAR REQA, REQB
                                                MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
                                                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
                                                BEQ     7$              ;BRANCH IF EQUAL
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #146,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        7$:     CLR     $GDDAT          ;EXPECT ZEROS
                                        ;REPORT ERROR IF RMEC2 NOT RESET
                                                MOV     RMEC2I,$BDDAT   ;CHECK RMEC2
                                                BEQ     8$              ;BRANCH IF 0
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #150,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        ;REPORT ERROR IF RMER2 NOT RESET
                                        8$:     MOV     RMER2I,$BDDAT   ;CHECK RMER2
                                                BEQ     9$              ;BRANCH IF NO ERROR
                                                ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
                                                MOVB    #147,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                                NOP
                                        9$:

                                        10$:
```

DRIVE CLEAR STATUS CHECK SUBROUTINE

```
115                                     ;AUGMENT RETURN ADDRESS IF ANY ERROR WAS FOUND
116 054136  062716  000004                     ADD     #4,(SP)          ;MOVE SP TO ERROR CALL
117 054142  105776  000000                     TSTB    @(SP)            ;WAS AN ERROR DETECTED??
118 054146  001403                             BEQ     11$              ;NO!!
119 054150  062716  000004                     ADD     #4,(SP)          ;YES - MOVE SP TO ERROR RETURN
120 054154  000402                             BR      12$
121 054156  162716  000004             11$:    SUB     #4,(SP) ;MOVE SP BACK TO NO ERROR RETURN
122 054162  000240                     12$:    NOP
123 054164  000207                             RTS     PC               ;RETURN TO USER
```

```
                                .SBTTL  DATA TRANSFER COMMAND STATUS CHECK SUBROUTINE

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

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

054166                          DTASTS:

                                ;CLEAR USER'S ERROR CALL AND ERROR FLAGS
054166  062716  000004                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
054172  105076  000000                CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF TRAP
054176  162716  000004                SUB     #4,(SP)         ;RESTORE SP TO NO ERROR
054202  005037  057602                CLR     49$             ;CLEAR ERROR FLAGS

                                ;REPORT ANY CONTROL BUS PARITY ERROR WHILE READING REMOTE REGISTERS,
                                ;I.E., MCPE = 1
054206  032737  020000  001336        BIT     #MCPE,RMCS1I    ;WAS THERE A PARITY ERROR??
054214  001422                      BEQ     1$              ;NO!!
054216  013737  001336  001140      MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
054224  042737  020000  001140      BIC     #MCPE,$GDDAT
054232  013737  001336  001142      MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
054240  062716  000004              ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054244  112776  000013  000000      MOVB    #13,@(SP)       ;WRITE ERROR NUMBER
054252  162716  000002              SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054256  004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054260  000466                      BR      4$
054262                          1$:

                                ;REPORT ANY CONTROL BUS PARITY ERROR WHILE WRITTING REMOTE REGISTERS,
                                ;I.E., PAR = 1 AND DPE = 0
054262  032737  000010  001352        BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
054270  001435                      BEQ     3$              ;NO!!
054272  032737  000010  001400      BIT     #DPE,RMER2I     ;DATA PARITY ERROR ?
054300  001031                      BNE     3$              ;YES!!
054302  013737  001352  001140      MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054310  042737  000010  001140      BIC     #PAR,$GDDAT
054316  013737  001352  001142      MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054324  062716  000004              ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
054330  112776  000050  000000      MOVB    #50,@(SP)       ;WRITE ERROR NUMBER
054336  032737  001000  001346      BIT     #MXF,RMCS2I     ;DID MXF GET SET??
054344  001003                      BNE     2$              ;YES!!
054346  112776  000274  000000      MOVB    #274,@(SP)      ;NO - CHANGE ERROR NUMBER
054354  162716  000002          2$: SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054360  004736                      JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054362  000425                      BR      4$

                                ;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
054364                                  3$:
054364  032737  001000  001346                BIT     #MXF,RMCS2I     ;WAS "MISSED TRANSFER" SET??
054372  001425                                BEQ     5$              ;NO!!
054374  013737  001346  001140                MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
054402  042737  001000  001140                BIC     #MXF,$GDDAT
054410  013737  001346  001142                MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
054416  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054422  112776  000275  000000                MOVB    #275,@(SP)      ;WRITE ERROR NUMBER
054430  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054434  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054436                                  4$:

                                        ;RESTORE SP TO NO ERROR RETURN AND BYPASS FURHTER STATUS CHECKING
054436  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
054442  000137  057554                        JMP     46$             ;SKIP TO END OF SUB

054446                                  5$:

                                        ;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"
054446  032737  020000  001352                BIT     #OPI,RMER1I     ;IS "OPI" SET??
054454  001447                                BEQ     8$              ;NO!!
054456  013737  001352  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
054464  042737  020000  001140                BIC     #OPI,$GDDAT
054472  013737  001352  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
054500  032737  010000  001350                BIT     #MOL,RMDSI      ;WAS MEDIUM OFF LINE??
054506  001404                                BEQ     6$              ;YES!!
054510  032737  000100  001350                BIT     #VV,RMDSI       ;WAS "MOL" INTERMITTENT??
054516  001013                                BNE     7$              ;NO!!
054520  062716  000004                6$:     ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054524  112776  000276  000000                MOVB    #276,@(SP)      ;WRITE ERROR NUMBER IN CALL
054532  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054536  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054540  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
054544  000413                                BR      8$
054546                                  7$:

                                        ;REPORT "OPI" ERROR, WHICH IS DUE TO "ON CYLINDER" NOT DROPPING OR
                                        ;"RUN" TIMEOUT (20 MS) OR SEARCH TIMEOUT (50 MS)
054546  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
054552  112776  000277  000000                MOVB    #277,@(SP)      ;WRITE ERROR NUMBER IN CALL
054560  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
054564  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
054566  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
054572  000240                                NOP
054574                                  8$:

                                        ;LOOK FOR "IVC" ERROR DURING COMMAND INITIATION
054574  032737  010000  001400                BIT     #IVC,RMER2I     ;WAS THERE AN "IVC" ERROR??
054602  001432                                BEQ     10$             ;NO!!
                                        ;REPORT "IVC" ERROR DUE TO "VV" = 0, OR REPORT ERRONEOUS "IVC" ERROR
054604  013737  001400  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
```

```
                        BIC     #IVC,$GDDAT
                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                        MOVB    #300,@(SP)      ;WRITE ERROR NUMBER IN CALL
                        BIT     #VV,RMDSI       ;WAS VOLUME VALID??
                        BEQ     9$              ;NO!!
                        MOVB    #301,@(SP)      ;CHANGE ERROR NUMBER
9$:                     SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT "IVC" ERROR AND RETURN
                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
10$:

;SEE IF "ILF" OR "RMR" IS SET
                        BIT     #ILR!ILF!RMR, RMER1I
                        BEQ     13$             ;NO ERRORS DETECTED
;REPORT AN ERROR IF "ILR" IS SET
                        BIT     #ILR,RMER1I     ;WAS "ILR" DETECTED??
                        BEQ     11$             ;NO!!
                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                        BIC     #ILR,$GDDAT
                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                        MOVB    #302,@(SP)      ;WRITE ERROR NUMBER IN CALL
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                        NOP
11$:

;REPORT AN ERROR IF "ILF" IS SET
                        BIT     #ILF,RMER1I     ;WAS "ILF" DETECTED??
                        BEQ     12$             ;NO!!
                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                        BIC     #ILF,$GDDAT
                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                        MOVB    #303,@(SP)      ;WRITE ERROR NUMBER IN CALL
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERORR AND RETURN
                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                        NOP
12$:

;REPORT AN ERROR IF "RMR" IS SET
                        BIT     #RMR,RMER1I     ;WAS "RMR" DETECTED??
                        BEQ     13$             ;NO!!
                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                        BIC     #RMR,$GDDAT
                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                        MOVB    #304,@(SP)      ;WRITE ERROR NUMBER IN CALL
                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                        NOP
13$:
;DETERMINE WHETHER OR NOT "IAE" SHOULD BE SET AND CHECK FOR ERROR
```

```
                                                MOV     #IAE,$GDDAT     ;SETUP FOR "IAE" = 1
                                                BIS     #SKI,49$        ;SETUP FOR "SKI" = 1
                                                CMP     RMDC0,#560.     ;GREATER THAN LAST CYLINDER ?
                                                BHI     15$             ;YES - CYLINDER IS INVALID
                                                BIC     #SKI,49$        ;RESET SKI FLAG

                                                CMPB    RMDA0+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
                                                BHI     15$             ;YES - TRACK IS INVALID

                                                CMPB    RMDA0,#29.      ;IS SECTOR > 29. ?
                                                BLOS    14$             ;NO
                                                BIT     #FMT16,RMOF0    ;18 BIT FORMAT ?
                                                BEQ     15$             ;YES - SECTOR IS INVALID FOR 18 BIT MODE
                                                CMPB    RMDA0,#30.      ;IS SECTOR > 30. ?
                                                BLOS    14$             ;NO
                                                BIT     #SSEI,RMOF0     ;IS SSEI CLEAR ?
                                                BEQ     15$             ;YES - SECTOR IS INVALID FOR 16 BIT MODE
                                                CMPB    RMDA0,#31.      ;IS SECTOR > 31. ?
                                                BHI     15$             ;YES - SECTOR IS INVALID
                                        14$:    CLR     $GDDAT          ;"IAE" SHOULD = 0

                                        15$:    MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     #^CIAE,$BDDAT
                                                CMP     $GDDAT,$BDDAT   ;IS "IAE" STATUS OK??
                                                BNE     16$             ;NO!!
                                                BIC     #SKI,49$        ;IAE OK - SKI SHOULD BE 0
                                                BR      17$
                                        16$:    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #305,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        17$:

                                        ;REPORT AN ERROR IF "SKI" IS SET AND "IAE" STATUS WAS OK
                                                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                                BIC     #^CSKI,$BDDAT
                                                MOV     49$,$GDDAT      ;EXPECTED STATUS
                                                BIC     #^CSKI,$GDDAT
                                                BIT     #SKI,RMER2I     ;WAS "SKI" SET??
                                                BEQ     18$             ;NO!!
                                                BIT     #SKI,49$        ;WAS SKI CAUSED BY IAE = 0??
                                                BNE     19$             ;YES - DON'T REPORT SKI
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #306,@(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
                                                BR      19$

                                        18$:

                                        ;REPORT AN ERROR IF SKI = 0 AND IAE WAS NOT DETECTED
                                                BIT     #SKI,49$        ;SHOULD SKI BE SET??
                                                BEQ     19$             ;NO!!
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #307,@(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)        ;RESTORE SP TO NO ERROR
                                        NOP
                                19$:

                                ;LOOK FOR "LSC" OR "LBC" OR "DVC" IN ERROR REGISTER #2
                                        BIT     #LSC!LBC!DVC,RMER2I
                                        BEQ     22$             ;NO ERRORS SET

                                ;REPORT ANY DEVICE FAULT, I.E., "DVC" = 1
                                        BIT     #DVC,RMER2I     ;IS "DVC" = 1??
                                        BEQ     20$             ;NO!!
                                        MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #DVC,$GDDAT
                                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
                                        MOVB    #310,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        NOP
                                20$:

                                ;REPORT LOSS OF BIT CLOCK, I.E.; "LBC" = 1, IF "MOL" = 1
                                        BIT     #LBC,RMER2I     ;IS LBC SET??
                                        BEQ     21$             ;NO!!
                                        BIT     #MOL,RMDSI      ;WAS LBC ERROR BY MOL = 0
                                        BEQ     21$             ;YES!!
                                        MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #LBC,$GDDAT
                                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #311,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        NOP
                                21$:

                                ;REPORT LOS OF SYSTEM CLOCK, I.E., "LSC" = 1
                                        BIT     #LSC,RMER2I     ;IS "LSC" = 1??
                                        BEQ     22$             ;NO!!
                                        MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #LSC,$GDDAT
                                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #312,@(SP)      ;WRITE ERROR NUMBER
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        NOP
                                22$:

                                ;LOOK FOR "UNS" OR "DTE" OR "WLE" IN ERROR REGISTER #1
                                        BIT     #UNS!DTE!WLE,RMER1I
                                        BEQ     27$             ;NO BITS SET
                                ;REPORT "UNS" ERROR IF "DVC" = 0
```

```
286 055716  032737  040000  001352          BIT     #UNS,RMER1I     ;IS 'UNS' SET??
287 055724  001427                          BEQ     23$             ;NO!!
288 055726  032737  000200  001400          BIT     #DVC,RMER2I     ;WAS 'UNS' CAUSED BY 'DVC'??
289 055734  001023                          BNE     23$             ;YES!!
290 055736  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
291 055744  042737  040000  001140          BIC     #UNS,$GDDAT
292 055752  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
293 055760  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
294 055764  112776  000313  000000          MOVB    #313,@(SP)      ;WRITE ERROR NUMBER
295 055772  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
296 055776  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
297 056000  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
298 056004                          23$:
299
300                                 ;REPORT ANY DRIVE TIMING ERROR, I.E., 'DTE" = 1
301 056004  032737  010000  001352          BIT     #DTE,RMER1I     ;IS DTE SET??
302 056012  001423                          BEQ     24$             ;NO!!
303 056014  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
304 056022  042737  010000  001140          BIC     #DTE,$GDDAT
305 056030  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
306 056036  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
307 056042  112776  000314  000000          MOVB    #314,@(SP)      ;WRITE ERROR NUMBER IN CALL
308 056050  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
309 056054  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
310 056056  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
311 056062                          24$:
312
313                                 ;REPORT AN ERROR IF WRITE LOCK ERROR IS SET.  SEE IF DRIVE IS NOT
314                                 ;WRITE PROTECTED, OR IF FUNCTION WAS NOT A WRITE
315 056062  032737  004000  001352          BIT     #WLE,RMER1I     ;WAS 'WLE" SET??
316 056070  001441                          BEQ     27$             ;NO!!
317 056072  013737  001352  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
318 056100  013737  001352  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
319 056106  052737  004000  001140          BIS     #WLE,$GDDAT
320 056114  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
321 056120  112776  000315  000000          MOVB    #315,@(SP)      ;WRITE ERROR NUMBER IN CALL
322 056126  032737  004000  001350          BIT     #WRL,RMDSI      ;WAS DRIVE WRITE PROTECTED??
323 056134  001404                          BEQ     25$             ;NO!!
324 056136  032737  000010  001412          BIT     #BIT3,RMCS1O    ;WAS COMMAND A WRITE??
325 056144  001406                          BEQ     26$             ;YES!!
326 056146  112776  000316  000000  25$:    MOVB    #316,@(SP)      ;CHANGE ERROR NUMBER
327 056154  042737  004000  001140          BIC     #WLE,$GDDAT
328 056162  162716  000002          26$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
329 056166  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
330 056170  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
331
332 056174                          27$:
333
334                                 ;OMIT DATA ERROR CHECKS IF ANY PREVIOUS ERRORS HAVE BEEN DETECTED
335 056174  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
336 056200  105776  000000                  TSTB    @(SP)           ;WAS ERROR DETECTED??
337 056204  001404                          BEQ     28$             ;NO - DO DATA CHECKS
338 056206  162716  000004                  SUB     #4,(SP)         ;RESTORE SP
339 056212  000137  057214                  JMP     42$             ;SKIP DATA CHECKS
340 056216  162716  000004          28$:    SUB     #4,(SP)         ;RESTORE SP
341
342                                 ;CHECK HEADER ERRORS IF FUNCTION WAS NOT WRITE HEADER AND DATA, AND
```

```
                                        ;IF HEADER COMPARE IS NOT INHIBITED
056222  013737  001412  057604            MOV     RMCS10,50$      ;STRIP AND STORE FUNCTION CODE
056230  042737  177700  057604            BIC     #^CFNCMSK,50$
056236  022737  000063  057604            CMP     #WH!GO,50$      ;WAS FUNCTION WRITE HEADER & DATA??
056244  001512                            BEQ     31$             ;YES - SKIP HEADER CHECKS
056246  032737  002000  001370            BIT     #HCI,RMOFI      ;WAS HCI SET??
056254  001106                            BNE     31$             ;YES - SKIP HEADER CHECKS

                                        ;SEE IF ANY HEADER ERRORS ARE SET, I.E., "FER" OR "HCRC" OR "HCE"
056256  032737  000620  001352            BIT     #HCRC!FER!HCE,RMER1I
056264  001533                            BEQ     33$             ;NO ERRORS SET

                                        ;REPORT HEADER CRC ERROR IF SET
056266  032737  000400  001352            BIT     #HCRC,RMER1I    ;WAS HCRC SET??
056274  001422                            BEQ     29$             ;NO!!
056276  013737  001352  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056304  042737  000400  001140            BIC     #HCRC,$GDDAT
056312  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056320  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
056324  112776  000317  000000            MOVB    #317,@(SP)      ;WRITE ERROR NUMBER
056332  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056336  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056340  000501                            BR      32$
056342                                  29$:

                                        ;REPORT FORMAT ERROR IF SET
056342  032737  000020  001352            BIT     #FER,RMER1I     ;WAS "FER" SET??
056350  001422                            BEQ     30$             ;NO!!
056352  013737  001352  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056360  042737  000020  001140            BIC     #FER,$GDDAT
056366  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056374  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USERS ERROR
056400  112776  000320  000000            MOVB    #320,@(SP)      ;WRITE ERROR NUMBER
056406  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056412  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056414  000453                            BR      32$
056416                                  30$:

                                        ;REPORT HEADER COMPARE ERROR IF SET
056416  032737  000200  001352            BIT     #HCE,RMER1I     ;WAS "HCE" SET??
056424  001453                            BEQ     33$             ;NO!!
056426  013737  001352  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056434  042737  000200  001140            BIC     #HCE,$GDDAT
056442  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056450  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
056454  112776  000321  000000            MOVB    #321,@(SP)      ;WRITE ERROR NUMBER
056462  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
056466  004736                            JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
056470  000425                            BR      32$
                                        ;THERE SHOULD BE NO HEADER ERRORS BECAUSE
                                        ;       .COMMAND WAS WRITE HEADER AND DATA, OR
                                        ;       .HEADER COMPARE INHIBIT WAS SET
056472  032737  000620  001352        31$:BIT     #HCE!FER!HCRC,RMER1I
056500  001425                            BEQ     33$             ;NO ERRORS WERE SET
056502  013737  001352  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056510  042737  000620  001140            BIC     #HCE!FER!HCRC,$GDDAT
056516  013737  001352  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

```
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #322,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        32$:    SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                                JMP     42$             ;OMIT FURTHER DATA CHECKS

                                        33$:

                                        ;IF COMMAND WAS A WRITE COMMAND, GO DO WRITE ERROR CHECKS, OTHERWISE
                                        ;DO READ ERROR CHECKS
                                                BIT     #BIT3,50$       ;WAS THIS A WRITE COMMAND?
                                                BNE     34$             ;NO!!
                                                JMP     39$             ;GO DO WRITE STATUS CHECK
                                        34$:

                                        ;REPORT DATA CHECK IF SET
                                                BIT     #DCK,RMER1I     ;DATA CHECK ERROR??
                                                BEQ     37$             ;NO!!
                                                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #DCK,$GDDAT
                                                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                                MOVB    #323,@(SP)      ;WRITE ERROR NUMBER
                                                BIT     #ECI,RMOFI      ;WAS ECC CORRECTION DISABLED??
                                                BNE     36$             ;YES!!
                                                MOVB    #324,@(SP)      ;CHANGE TO RECOVERABLE ERROR
                                                BIT     #ECH,RMER1I     ;IS ERROR RECOVERABLE??
                                                BNE     35$             ;NO !!
                                        ;DO NOT REPORT RECOVERABLE ERROR IF READ COMMAND
                                                BIT     #BIT4,50$       ;WAS THIS A READ COMMAND ??
                                                BEQ     36$             ;NO !!
                                                SUB     #4,(SP)         ;RESTORE SP
                                                BR      37$             ;SKIP ERROR - DATA WILL BE CORRECTED
                                        35$:    MOVB    #325,@(SP)      ;CHANGE TO NON RECOVERABLE
                                        36$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR

                                        37$:

                                        ;REPORT DATA BUS PARITY ERROR IF SET, I.E., MDPE = 1
                                                BIT     #MDPE,RMCS2I    ;PARITY ERROR SET??
                                                BEQ     38$             ;NO!!
                                                MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
                                                BIC     #MDPE,$GDDAT
                                                MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                                MOVB    #326,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        38$:    JMP     42$             ;SKIP WRITE STATUS CHECK

                                        39$:

                                        ;TEST TO SEE THAT OFFSET MODE WAS RESET; REPORT ERROR IF "OM" = 1
```

```
                                        BIT     #OM,RMDSI       ;IS OFFSET ON??
                                        BEQ     40$             ;NO
                                        MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                        BIC     #OM,$GDDAT
                                        MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #327,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                40$:

                                ;TEST FOR DATA BUS PARITY ERROR; REPORT ERROR IF 'DPE'' = 1
                                        BIT     #DPE,RMER2I     ;DATA PARITY ERROR??
                                        BEQ     41$             ;NO!!
                                        MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #DPE,$GDDAT
                                        MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #330,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                41$:

                                ;TEST FOR WRITE CLOCK FAILURE; REPORT ERROR IF 'WCF'' = 1
                                        BIT     #WCF,RMER1I     ;IS 'WCF'' SET??
                                        BEQ     42$             ;NO!!
                                        MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #WCF,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #331,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                42$:

                                ;REPORT 'DATA LATE'' ERROR IF 'DLT'' = 1
                                        BIT     #DLT,RMCS2I     ;IS 'DLT'' SET??
                                        BEQ     43$             ;NO!!
                                        MOV     RMCS2I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #DLT,$GDDAT
                                        MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #332,@(SP)      ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                43$:
                                ;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     46$             ;YES!!

                                ;REPORT ERROR IF POSITIONING IN PROGRESS AND NO SEEK INCOMPLETE ERROR,
```

```
                                        ;I.E. PIP = 1 AND SKI = 0
057310  032737  020000  001350                BIT     #PIP,RMDSI      ;IS "PIP" SET??
057316  001430                                BEQ     44$             ;NO!!
057320  032737  040000  001400                BIT     #SKI,RMER2I     ;WAS "SKI" ERROR REPORTED??
057326  001024                                BNE     44$             ;YES-DONT REPORT PIP
057330  013737  001350  001140                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
057336  042737  020000  001140                BIC     #PIP,$GDDAT
057344  013737  001350  001142                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
057352  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
057356  112776  000333  000000                MOVB    #333,@(SP)      ;WRITE ERROR NUMBER
057364  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057370  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057372  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
057376  000240                                NOP
057400                                  44$:

                                        ;REPORT ERROR IF MEDIUM IS NOT ON LINE AND OPI ERROR WAS NOT
                                        ;REPORTED, I.E., MOL = OPI = 0
057400  032737  010000  001350                BIT     #MOL,RMDSI      ;IS MEDIUM ON LINE??
057406  001027                                BNE     45$             ;YES!!
057410  032737  020000  001352                BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
057416  001023                                BNE     45$             ;YES!!
057420  013737  001350  001140                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
057426  052737  010000  001140                BIS     #MOL,$GDDAT
057434  013737  001350  001142                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
057442  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
057446  112776  000334  000000                MOVB    #334,@(SP)      ;WRITE ERROR NUMBER
057454  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057460  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057462  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
057466                                  45$:

                                        ;REPORT ERROR IF VOLUME IS NOT VALID AND "IVC" ERROR WAS NOT
                                        ;REPORTED, I.E., VV = IVC = 0
057466  032737  000100  001350                BIT     #VV,RMDSI       ;IS VOLUME VALID??
057474  001027                                BNE     46$             ;YES!!
057476  032737  010000  001400                BIT     #IVC,RMER2I     ;WAS IVC ERROR REPORTED??
057504  001033                                BNE     47$             ;YES!!
057506  013737  001350  001140                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
057514  052737  000100  001140                BIS     #VV,$GDDAT
057522  013737  001350  001142                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
057530  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
057534  112776  000335  000000                MOVB    #335,@(SP)      ;WRITE ERROR NUMBER
057542  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
057546  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
057550  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
057554                                  46$:

                                        ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS FOUND
057554  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
057560  105776  000000                        TSTB    @(SP)           ;ANY ERROR??
057564  001403                                BEQ     47$             ;NO!!
057566  062716  000004                        ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
057572  000402                                BR      48$
057574  162716  000004              47$:      SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN

057600  000207                      48$:      RTS     PC              ;RETURN TO USER
```

```
    571
    572 057602  000000                49$:    .WORD                   ;ERROR FLAGS
    573 057604  000000                50$:    .WORD                   ;TEMPORARY STORAGE
```

```
                                .SBTTL  STATIC DRIVE STATUS CHECK SUBROUTINE

                                ;THIS SUBROUTINE LOOKS FOR UNEXPECTED CHANGES IN DRIVE
                                ;STATUS, SUCH AS THE DRIVE LOSING VOLUME VALID.  THE SUBROUTINE
                                ;CAN BE USED BY HOUSEKEEPING AND OTHER COMMANDS DURING WHICH THERE
                                ;SHOULD NOT BE ANY DRIVE ERRORS OR CHANGES IN STATE.

                                ;THE FOLLOWING CONDITIONS ARE TESTED AND REPORTED AS ERRORS
                                ;IF TRUE:

                                ;       .MOL = 0, INDICATES DRIVE WENT OFFLINE, NOTE
                                ;THAT MOL IS ASSUMED TO HAVE BEEN SET
                                ;       .VV = 0, INDICATES THE DRIVE LOST VOLUME VALID
                                ;       .PIP = 1, INDICATES THAT THE DRIVE IS OFF CYLINDER
                                ;       .SKI = 1, INDICATES THE DRIVE HAS AN UNEXPECTED SKI ERROR
                                ;       .DVC = 1, INDICATES AN UNEXPECTED DEVICE FAULT

                                ;THE SUBROUTINE IS CALLED AFTER STORING STATUS IN THE GET BUFFER.

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

057606                          STCDRVSTS:

                                ;CLEAR USER'S ERROR CALL
057606  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
057612  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
057616  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR RETURN
                                ;SEE IF 'MOL" = 'VV" = 1, AND 'PIP" = 0
057622  013746  001350                  MOV     RMDSI,-(SP)     ;PUT DRIVE STATUS ON STACK
057626  042716  147677                  BIC     #^C<PIP!MOL!VV>,(SP)
057632  022726  010100                  CMP     #MOL!VV,(SP)+   ;ARE MOL,VV AND PIP O.K.??
057636  001524                          BEQ     3$              ;YES!!

                                ;REPORT AN ERROR IF MOL = 0 AND 'OPI" = 0
057640  032737  010000  001350          BIT     #MOL,RMDSI      ;IS MOL ON ??
057646  001030                          BNE     1$              ;YES!!
057650  032737  020000  001352          BIT     #OPI,RMER1I     ;WAS "OPI" SET??
057656  001024                          BNE     1$              ;YES-DONT REPORT 'MOL" = 0
057660  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
057666  052737  010000  001140          BIS     #MOL,$GDDAT
057674  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
057702  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057706  112776  000207  000000          MOVB    #207,@(SP)      ;WRITE ERROR NUMBER IN CALL
057714  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057720  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
057722  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
057726  000240                          NOP
057730                          1$:

                                ;REPORT AN ERROR IF VOLUME VALID IS NOW ZERO AND "IVC" = 0
057730  032737  000100  001350          BIT     #VV,RMDSI       ;IS 'VV" = 0??
057736  001030                          BNE     2$              ;NO!!
```

```
     058 057740  032737  010000  001400          BIT     #IVC,RMER2I     ;WAS "IVC" SET??
     059 057746  001024                          BNE     2$              ;YES-DONT REPORT "VV" = 0
     060 057750  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
     061 057756  052737  000100  001350          BIS     #VV,RMDSI
     062 057764  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
     063 057772  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     064 057776  112776  000210  000000          MOVB    #210,@(SP)      ;WRITE ERROR NUMBER IN CALL
     065 060004  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     066 060010  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     067 060012  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     068 060016  000240                          NOP
     069 060020                          2$:
     070
     071                                 ;REPORT AN ERROR IF DRIVE IS OFF CYLINDER AND "SKI" = 0
     072 060020  032737  020000  001350          BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
     073 060026  001430                          BEQ     3$              ;NO!!
     074 060030  032737  040000  001400          BIT     #SKI,RMER2I     ;WAS "SKI" SET?!
     075 060036  001024                          BNE     3$              ;YES-DONT REPORT "PIP" = 1
     076 060040  013737  001350  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
     077 060046  042737  020000  001140          BIC     #PIP,$GDDAT
     078 060054  013737  001350  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
     079 060062  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     080 060066  112776  000211  000000          MOVB    #211,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
     081 060074  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     082 060100  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     083 060102  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     084 060106  000240                          NOP
     085 060110                          3$:
     086
     087                                 ;SEE IF "SKI" = "DVC" = 0
     088 060110  013746  001400                  MOV     RMER2I,-(SP)    ;PUT ERROR REG 2 ON STACK
     089 060114  042726  137577                  BIC     #^C<SKI!DVC>,(SP)+
     090 060120  001460                          BEQ     6$              ;BRANCH IF NO ERROR
     091 060122                          4$:
     092
     093                                 ;REPORT AN ERROR IF THERE IS A DEVICE FAULT
     094 060122  032737  000200  001400          BIT     #DVC,RMER2I     ;ANY DEVICE FAULT??
     095 060130  001424                          BEQ     5$              ;NO!!
     096 060132  013737  001400  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
     097 060140  042737  000200  001140          BIC     #DVC,$GDDAT
     098 060146  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     099 060154  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S CALL
     100 060160  112776  000212  000000          MOVB    #212,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
     101 060166  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     102 060172  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     103 060174  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     104 060200  000240                          NOP
     105 060202                          5$:
     106
     107                                 ;REPORT AN ERROR IF "SKI" = 1
     108 060202  032737  040000  001400          BIT     #SKI,RMER2I     ;IS THERE A SEEK INCOMPLETE ERROR
     109 060210  001424                          BEQ     6$              ;NO!!
     110 060212  013737  001400  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
     111 060220  042737  040000  001140          BIC     #SKI,$GDDAT
     112 060226  013737  001400  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     113 060234  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     114 060240  112776  000213  000000          MOVB    #213,@(SP)      ;WRITE ERROR NUMBER IN USER'S ERROR CALL
```

```
    115 060246  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    116 060252  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    117 060254  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    118 060260  000240                          NOP
    119 060262                          6$:
    120
    121                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
    122 060262  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    123 060266  105776  000000                  TSTB    @(SP)           ;WAS AN ERROR DETECTED??
    124 060272  001403                          BEQ     7$              ;NO!!
    125 060274  062716  000004                  ADD     #4,(SP)         ;YES - MOVE SP TO USER'S ERROR RETURN
    126 060300  000402                          BR      8$
    127 060302  162716  000004          7$:     SUB     #4,(SP)         ;NO - MOVE SP TO NO ERROR RETURN
    128 060306  000240                  8$:     NOP
    129 060310  000207                          RTS     PC              ;RETURN TO USER
```

```
SAVE AND RESTORE R0-R5 ROUTINES

      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

     060312                             $SAVREG:
     060312  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
     060314  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
     060316  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
     060320  010346                             MOV     R3,-(SP)        ;;PUSH R3 ON STACK
     060322  010446                             MOV     R4,-(SP)        ;;PUSH R4 ON STACK
     060324  010546                             MOV     R5,-(SP)        ;;PUSH R5 ON STACK
     060326  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
     060332  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
     060336  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
     060342  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
     060346  000002                             RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
     060350                             $RESREG:
     060350  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
     060354  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
     060360  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
     060364  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
     060370  012605                             MOV     (SP)+,R5        ;;POP STACK INTO R5
     060372  012604                             MOV     (SP)+,R4        ;;POP STACK INTO R4
     060374  012603                             MOV     (SP)+,R3        ;;POP STACK INTO R3
     060376  012602                             MOV     (SP)+,R2        ;;POP STACK INTO R2
     060400  012601                             MOV     (SP)+,R1        ;;POP STACK INTO R1
     060402  012600                             MOV     (SP)+,R0        ;;POP STACK INTO R0
     060404  000002                             RTI
```

```
                                        .SBTTL  BINARY TO ASCII AND TYPE ROUTINE

                                        ;;*****************************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
                                        ;*BINARY-ASCII NUMBER AND TYPE IT.
                                        ;*CALL:
                                        ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
                                        ;*      TYPBN                   ;;TYPE IT

060406  010146                  $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
060410  016601  000006                  MOV     6(SP),R1        ;;GET THE INPUT NUMBER
060414  000261                          SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
060416  112737  000060  060460  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
060424  006101                          ROL     R1              ;;GET THIS BIT
060426  001406                          BEQ     2$              ;;DONE?
060430  105537  060460                  ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
060434  104401  060460                  TYPE    ,$BIN           ;;GO TYPE THIS BIT
060440  000241                          CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
060442  000765                          BR      1$              ;;GO DO THE NEXT BIT
060444  012601                  2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
060446  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
060454  012616                          MOV     (SP)+,(SP)
060456  000002                          RTI                     ;;RETURN TO USER
060460     000     000          $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;*****************************************************************
                                ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;;*REPLACED WITH SPACES.
                                ;;*CALL:
                                ;;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;;*      TYPDS                   ;;GO TO THE ROUTINE

060462                          $TYPDS:
060462  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
060464  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
060466  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
060470  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
060472  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
060474  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
060500  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
060504  100004                          BPL     1$              ;;BR IF INPUT IS POS.
060506  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
060510  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
060516  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
060520  012703  060676                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
060524  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
060530  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
060532  016001  060666                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
060536  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
060540  002402                          BLT     4$              ;;BR IF DONE
060542  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
060544  000774                          BR      3$
060546  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
060550  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
060552  001002                          BNE     5$              ;;FALL THROUGH IF 0
060554  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
060556  100407                          BMI     7$              ;;BR IF YES
060560  106316                  5$:     ASLB    (SP)            ;;MSD?
060562  103003                          BCC     6$              ;;BR IF NO
060564  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
060572  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
060576  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
060602  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
060604  005720                          TST     (R0)+           ;;JUST INCREMENTING
060606  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
060612  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
060614  003002                          BGT     8$              ;;GO TO EXIT
060616  010502                          MOV     R5,R2           ;;GET THE LSD
060620  000764                          BR      6$              ;;GO CHANGE TO ASCII
060622  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
060624  100003                          BPL     9$              ;;BR IF NO
060626  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
060634  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
060636  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
060640  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
060642  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
060644  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```
060646  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
060650  104401  060676                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
060654  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
060662  012616                          MOV     (SP)+,(SP)
060664  000002                          RTI                     ;;RETURN TO USER
060666  023420                  $DTBL:  10000.
060670  001750                          1000.
060672  000144                          100.
060674  000012                          10.
060676                          $DBLK:  .BLKW   4
```

```
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

060706  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
060712  116637  000001  061131          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
060720  112637  061133                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
060724  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
060730  000406                          BR      $TYPON
060732  112737  000001  061131  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
060740  112737  000006  061133          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
060746  112737  000005  061130  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
060754  010346                          MOV     R3,-(SP)        ;;SAVE R3
060756  010446                          MOV     R4,-(SP)        ;;SAVE R4
060760  010546                          MOV     R5,-(SP)        ;;SAVE R5
060762  113704  061133                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
060766  005404                          NEG     R4
060770  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
060774  110437  061132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
061000  113704  061131                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
061004  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
061010  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
061012  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
061014  000404                          BR      3$              ;;GO DO MSB
061016  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
061020  006105                          ROL     R5
061022  006105                          ROL     R5
061024  010503                          MOV     R5,R3
061026  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
061030  105337  061132                  DECB    $OMODE          ;;TYPE THIS DIGIT?
061034  100016                          BPL     7$              ;;BR IF
061036  042703  177770                  BIC     #177770,R3      ;;GET R   OF JUNK
061042  001002                          BNE     4$              ;;TEST FOR 0
061044  005704                          TST     R4              ;;SUPPRESS THIS 0?
061046  001403                          BEQ     5$              ;;BR IF YES
061050  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
```

BINARY TO OCTAL (ASCII) AND TYPE

```
061052  052703  000060            BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
061056  052703  000040     5$:    BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
061062  110337  061126            MOVB    R3,8$           ;;SAVE FOR TYPING
061066  104401  061126            TYPE    ,8$             ;;GO TYPE THIS DIGIT
061072  105337  061130     7$:    DECB    $OCNT           ;;COUNT BY 1
061076  003347                    BGT     2$              ;;BR IF MORE TO DO
061100  002402                    BLT     6$              ;;BR IF DONE
061102  005204                    INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
061104  000744                    BR      2$              ;;GO DO THE LAST DIGIT
061106  012605             6$:    MOV     (SP)+,R5        ;;RESTORE R5
061110  012604                    MOV     (SP)+,R4        ;;RESTORE R4
061112  012603                    MOV     (SP)+,R3        ;;RESTORE R3
061114  016666  000002  000004    MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
061122  012616                    MOV     (SP)+,(SP)
061124  000002                    RTI                     ;;RETURN
061126     000     8$:    .BYTE   0               ;;STORAGE FOR ASCII DIGIT
061127     000            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
061130     000     $OCNT: .BYTE   0               ;;OCTAL DIGIT COUNTER
061131     000     $OFILL:.BYTE   0               ;;ZERO FILL SWITCH
061132  000000     $OMODE:.WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .SBTTL  TYPE ROUTINE

                                ;;******************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*       TYPE
                                ;*       MESADR
                                ;*

061134  105737  001173          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
061140  100002                  BPL     1$              ;;BR IF YES
061142  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
061144  000430                  BR      3$              ;;LEAVE
061146  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
061150  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
061154  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
061162  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
061164  132737  000100  001243          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
061172  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
061174  010037  061204                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
061200  004737  065546                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
061204  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
061206  132737  000040  001243  62$:     BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
061214  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
061216  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
061220  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
061222  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
061224  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
061226  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
061232  000002                  RTI                     ;;RETURN
061234  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
061240  001430                  BEQ     8$
061242  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
061246  001006                  BNE     5$
061250  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
061252  104401                  TYPE                    ;;TYPE A CR AND LF
061254  001217                  $CRLF
061256  105037  061464                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
061262  000755                  BR      2$              ;;GET NEXT CHARACTER
061264  004737  061346          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
061270  123726  001172          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
061274  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
061276  013746  001170                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                        ;;AND THE NULL CHAR.
061302  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
061306  002770                  BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
061310  004737  061346                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
061314  105337  061464                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
061320  000770                  BR      7$              ;;LOOP
```

```
                                        ;HORIZONTAL TAB PROCESSOR

061322  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
061326  004737  061346          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
061332  132737  000007  061464          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
061340  001372                          BNE     9$              ;;TAB STOP
061342  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
061344  000724                          BR      2$              ;;GET NEXT CHARACTER
061346                          $TYPEC:
061346  105777  117606                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
061352  100022                          BPL     10$             ;;BR IF NOT
061354  017746  117602                  MOV     @$TKB,-(SP)     ;;GET CHAR
061360  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
061364  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
061370  001012                          BNE     102$            ;;BR IF NOT
061372                          101$:
061372  105777  117562                  TSTB    @$TKS           ;;WAIT FOR CHAR
061376  100375                          BPL     101$
061400  117716  117556                  MOVB    @$TKB,(SP)      ;;GET CHAR
061404  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
061410  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
061414  001366                          BNE     101$            ;;BR IF NOT
061416                          102$:
061416  005726                          TST     (SP)+           ;;FIX STACK
061420                          10$:
061420  105777  117540                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
061424  100375                          BPL     10$
061426  116677  000002  117532          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
061434  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
061442  001003                          BNE     1$              ;;BRANCH IF NO
061444  105037  061464                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
061450  000406                          BR      $TYPEX          ;;EXIT
061452  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
061460  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
061462  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
061464  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
061466  000207          $TYPEX: RTS     PC
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE

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

061470                          $SCOPE:
061470  104410                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
061472  004737  062222                 JSR     PC,STOP
061476  032777  040000  117450  1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
061504  001402                         BEQ     9$              ;;NO IF SW14=0
061506  000137  062136                 JMP     $OVER           ;;JUMP OVER SCOPE ROUTINE
061512                          9$:
                                ;#####START OF CODE FOR THE XOR TESTER#####
061512  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
061514  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
061520  012737  061540  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
061526  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
061532  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
061536  000561                         BR      $SVLAD          ;;GO TO THE NEXT TEST
061540  022626                  5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
061542  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
061546  000521                         BR      7$              ;;LOOP ON THE PRESENT TEST
061550                          6$:;#####END OF CODE FOR THE XOR TESTER#####
061550  032777  000400  117376         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
061556  001421                         BEQ     2$              ;;BR IF NO
061560  005046                         CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
061562  117716  117366                 MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
061566  001414                         BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
061570  022716  000023                 CMP     #23,(SP)        ;;CHECK THE NUMBER IN THE SWR
061574  002411                         BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
061576  011637  001116                 MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
061602  005316                         DEC     (SP)            ;;BACKUP BY ONE
061604  006316                         ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
061606  062716  062154                 ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
061612  013637  001122                 MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
061616  000547                         BR      $OVER           ;;GO LOOP ON THE TEST
061620  005726                  8$:    TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
061622  105737  001117          2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
061626  001502                         BEQ     3$              ;;BR IF NO
061630  022737  177777  062640         CMP     #-1,CPSAVE      ;;SEE IF TIMEOUT WAS PREVIOUSLY RECORDED
061636  001455                         BEQ     2003$           ;;KICK AROUND ROUTINE IF SO
061640  013746  000004                 MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
061644  012737  061662  000004         MOV     #2000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
061652  013737  177766  062640         MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
061660  000406                         BR      2001$
061662  012737  177777  062640  2000$: MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
061670  012716  061676                 MOV     #2001$,(SP)     ;;SETUP RETURN ADDRESS
```

```
061674  000002                          RTI
061676  012637  000004          2001$:  MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

061702  022737  177777  062640  2002$:  CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
061710  001430                          BEQ     2003$           ;;BRANCH IF SO
061712  032737  000001  062640          BIT     #BIT00,CPSAVE   ;;SEE IF THE POWER MONITOR BIT IS ON
061720  001424                          BEQ     2003$           ;;BRANCH TO CONTINUE ROUTINE IF CLEAR
061722  042737  000001  177766          BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND TO BE SET
061730  013746  001154                  MOV     SWR,-(SP)       ;;SAVE SWR ADDRESS
061734  017646  000000                  MOV     @(SP),-(SP)     ;;SAVE SWR VALUE
061740  012737  000176  001154          MOV     #176,SWR        ;;GET SOFTWARE SWR ADDRESS
061746  011677  117202                  MOV     (SP),@SWR       ;;GET CURRENT SWR VALUE
061752  042777  001000  117174          BIC     #BIT09,@SWR     ;;DON'T ALLOW LOOP ON ERROR ON THIS ERROR
061760  104177                          EMT     177             ;;CALL SPECIAL POWER FAIL BIT ERROR CALL
061762  012676  000000                  MOV     (SP)+,@(SP)     ;;RESTORE SWR TO ORIGINAL VALUE
061766  012637  001154                  MOV     (SP)+,SWR       ;;RESTORE SWR ADDRESS
061772                          2003$:
061772  123737  001131  001117          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
062000  101015                          BHI     3$              ;;BR IF NO
062002  032777  001000  117144          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
062010  001404                          BEQ     4$              ;;BR IF NO
062012  013737  001124  001122  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
062020  000446                          BR      $OVER
062022  105037  001117          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
062026  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
062032  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
062034  032777  004000  117112  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
062042  001011                          BNE     1$              ;;BR IF YES
062044  005737  001230                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
062050  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
062052  005237  001120                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
062056  023737  001206  001120          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
062064  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
062066  012737  000001  001120  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
062074  013737  062152  001206          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
062102  105237  001116          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
062106  113737  001116  001226          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
062114  011637  001122                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
062120  011637  001124                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
062124  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
062130  112737  000001  001131          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
062136  013777  001116  117012  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
062144  013716  001122                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
062150  000002                          RTI                     ;;FIXES PS
062152  000005                  $MXCNT: 5.                      ;;MAX. NUMBER OF ITERATIONS
062154                          $SW08TBL:
        000023                  .REPT   $TN-1
062154  007700                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
062156  010100                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
062160  011106                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
062162  012064                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
062164  013260                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
062166  014266                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
062170  015244                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
062172  016442                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
062174  017470                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
062176  020714                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
```

```
062200  021736                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
062202  022550                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
062204  024420                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
062206  025722                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
062210  026742                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
062212  027660                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
062214  030666                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
062216  031714                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
062220  033076                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
                                
                                ;DROP PRIORITY TO ALLOW CONSOLE INTERRUPT
                                
062222                          STOP:
062222  012746  000140                  MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
062226  012746  062234                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
062232  000002                          RTI                     ;;POP NEW PC AND PS
062234                          64$:
                                
                                ;RAISE PRIORITY TO INHIBIT CONSOLE INTERRUPT
                                
062234  012746  000240                  MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
062240  012746  062246                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
062244  000002                          RTI                     ;;POP NEW PC AND PS
062246                          65$:
062246  000207                          RTS     PC              ;RETURN
```

```
     1                                  .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

062250  105037  062642         $ERROR. CLRB    IBSAVE          ;;CLEAR THE ITEM BYTE SAVE LOCATION
062254  104410                 CKSWR           ;;TEST FOR CHANGE IN SOFT-SWR
062256  105237  001117         7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
062262  001775                 BEQ     7$      ;;DON'T LET THE FLAG GO TO ZERO
062264  013777  001116  116664         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
062272  032777  002000  116654         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
062300  001402                 BEQ     1$              ;;NO - SKIP
062302  104401  001212         TYPE    ,$BELL          ;;RING BELL
062306  005237  001126         1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
062312  011637  001132         MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
062316  162737  000002  001132         SUB     #2,$ERRPC
062324  117737  116602  001130         MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
062332  032777  001000  116614         BIT     #BIT09,@SWR     ;;SEE IF LOOP ON ERROR IS SET
062340  001060                 BNE     1004$           ;;BRANCH AROUND ROUTINE IF SO
062342  122737  000177  001130         CMPB    #177,$ITEMB     ;;SEE IF THIS IS THE POWER FAIL CALL
062350  001454                 BEQ     1004$           ;;BRANCH AROUND ROUTINE IF IT IS
062352  105737  062642         TSTB    IBSAVE          ;;SEE IF THIS IS THE 2ND ERROR CALL IN THIS ROUTINE
062356  001047                 BNE     1003$           ;;BRANCH IF SO
062360  022737  177777  062640         CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
062366  001445                 BEQ     1004$           ;;BRANCH IF SO
062370  013746  000004         MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
062374  012737  062412  000004         MOV     #1000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
062402  013737  177766  062640         MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
062410  000406                 BR      1001$
062412  012737  177777  062640 1000$:  MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
062420  012716  062426         MOV     #1001$,(SP)     ;;SETUP RETURN ADDRESS
062424  000002                 RTI
062426  012637  000004 1001$:  MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

062432  022737  177777  062640 1002$:  CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
062440  001420                 BEQ     1004$           ;;BRANCH IF SO
062442  032737  000001  062640         BIT     #BIT00,CPSAVE   ;;SEE IF POWER MONITOR BIT IS SET IN CPU ERR REG
062450  001414                 BEQ     1004$           ;;BRANCH IF OK
062452  042737  000001  177766         BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND SET
062460  113737  001130  062642         MOVB    $ITEMB,IBSAVE   ;;MAKE IBSAVE NON-ZERO FOR DUAL ERROR CALL
062466  112737  000177  001130         MOVB    #177,$ITEMB     ;;SET $ITEMB TO SPECIAL POWER FAIL POINTER
062474  000402                 BR      1004$           ;;BRANCH OVER IBSAVE CLEARING

062476  105037  062642 1003$:  CLRB    IBSAVE          ;;CLEAR IBSAVE SO 2ND TIME THROUGH EXITS
062502                 1004$:
062502  032777  020000  116444         BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
062510  001004                 BNE     20$             ;;SKIP TYPEOUTS
062512  004737  062644         JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
```

ERROR HANDLER ROUTINE

```
062516  104401  001217            TYPE    ,$CRLF
062522                      20$:
062522  122737  000001  001242    CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
062530  001007                    BNE     2$              ;;NO.SKIP APT ERROR REPORT
062532  113737  001130  062544    MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
062540  004737  065556            JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
062544     000          21$:     .BYTE   0
062545     000                   .BYTE   0
062546  000777          22$:     BR      22$             ;;APT ERROR LOOP
062550  105737  062642    2$:     TSTB    IBSAVE          ;;SEE IF IBSAVE IS LOADED
062554  001005                    BNE     3$              ;;BRANCH IF NOT - NO HALT ON PWR MON BIT ERROR
062556  005777  116372            TST     @SWR            ;;HALT ON ERROR
062562  100002                    BPL     3$              ;;SKIP IF CONTINUE
062564  000000                    HALT                    ;;HALT ON ERROR!
062566  104410                    CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
062570                      3$:
062570  032777  001000  116356    BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
062576  001402                    BEQ     4$              ;;BR IF NO
062600  013716  001124            MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
062604  005737  001210    4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
062610  001402                    BEQ     5$              ;;BR IF NONE
062612  013716  001210            MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
062616                      5$:
062616  022737  033756  000042    CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
062624  001001                    BNE     6$              ;;BRANCH IF NO
062626  000000                    HALT                    ;;YES
062630                      6$:
062630  105737  062642            TSTB    IBSAVE          ;;SEE IF ITEM BYTE SAVE LOCATION HAS AN ERROR CALL
062634  001210                    BNE     7$              ;;BRANCH BACK TO CALL ORIGINAL ERROR
062636  000002                    RTI                     ;;RETURN
062640  000000          CPSAVE:  .WORD   0               ;;LOCATION TO SAVE CPU ERROR REG CONTENTS
062642  000000          IBSAVE:  .WORD   0               ;;LOCATION TO SAVE ITEM BYTE
```

```
                                        .SBTTL  ERROR TYPEOUT ROUTINE

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

062644  104414                  ERRTYP: SAVREG
062646  032777  020000  116300          BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
062654  001402                          BEQ     1$              ;NO!!
062656  000137  063454                  JMP     27$             ;YES!!

                                        ;TYPE UNIT NUMBER, DRIVE TYPE, TEST NUMBER, ERROR NUMBER, AND
                                        ;PROGRAM COUNTER
062662  104401  001217          1$:     TYPE    ,$CRLF
062666  104401  063470                  TYPE    ,ERTY00         ;TYPE "DRV#"
062672  013746  001234                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                ;;TYPE DRIVE NUMBER
062676  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
062700     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
062701     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

                                        ;TYPE 'DRIVE TYPE', RM80 FOR UNIT UNDER TEST
062702  013700  001276                  MOV     $BASE,R0        ;GET RM BASE ADDRESS
062706  016000  000026                  MOV     RMDT(R0),R0     ;GET DRIVE TYPE REGISTER
062712  042700  177740                  BIC     #177740,R0      ;SAVE DRIVE TYPE BITS AND
062716  012737  067062  062740          MOV     #$RM80,3$       ;GET ASCII DRIVE TYPE
062724  022700  000026                  CMP     #26,R0          ;IS DEVICE AN RM80 ?
062730  001004                          BNE     4$              ;NO !!
062732  104401  063525          2$:     TYPE    ,ERTY05         ;TYPE " - "
062736  104401                          TYPE                    ;TYPE DRIVE TYPE
062740  000000          3$:     .WORD   0               ;DRIVE TYPE MESSAGE IS STORED HERE

                                        ;TYPE TEST NUMBER, ERROR NUMBER AND PROGRAM COUNTER
062742  005037  063460          4$:     CLR     TSTNMB          ;LOAD TEST NUMBER FOR
062746  013737  001226  063460          MOV     $TESTN,TSTNMB
062754  104401  063475                  TYPE    ,ERTY01         ;TYPE "TST#"
062760  013746  063460                  MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                ;;TYPE TEST NUMBER
062764  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
062766     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
062767     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
062770  005037  063462                  CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
062774  113737  001130  063462          MOVB    $ITEMB,ERRNMB   ;TYPEOUT
063002  001406                          BEQ     5$              ;SKIP IF NO ERROR CALLED
063004  104401  063505                  TYPE    ,ERTY02         ;TYPE "ERR#"
063010  013746  063462                  MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                ;;TYPE ERROR NUMBER
063014  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
063016     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
063017     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
063020  104401  063514          5$:     TYPE    ,ERTY03         ;TYPE "PC="
```

ERROR TYPEOUT ROUTINE

```
     46 063024  013746  001132                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                      ;;TYPE PROGRAM COUNTER
        063030  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
        063032     006                        .BYTE   6               ;;TYPE 6 DIGIT(S)
        063033     001                        .BYTE   1               ;;TYPE LEADING ZEROS
     47
     48                                ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
     49 063034  005737  063462        6$:     TST     ERRNMB          ;WAS AN ERROR CALLED?
     50 063040  001002                        BNE     7$              ;BR IF YES
     51 063042  000137  063454                JMP     27$             ;NO--EXIT
     52
     53 063046  104401  001217        7$:     TYPE    ,$CRLF          ;YES-TYPE CRLF
     54 063052  105037  063466                CLRB    BOTFLG          ;CLEAR BOT FLAG
     55 063056  105037  063467                CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
     56 063062  013700  063462                MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
     57 063066  122700  000177                CMPB    #177,R0         ;SEE IF THIS ERROR CALL IS SPECIAL POWER FAIL CALL
     58 063072  001003                        BNE     8$              ;BRANCH IF NOT
     59 063074  012700  063532                MOV     #PFECH,R0       ;MOVE POWER FAIL ERROR CALL TABLE TO R0
     60 063100  000405                        BR      9$
     61 063102  006300                8$:     ASL     R0              ;FOUR ENTRIES IN ERROR
     62 063104  006300                        ASL     R0              ;TABLE
     63 063106  006300                        ASL     R0
     64 063110  062700  001574                ADD     #$ERRTB-8.,R0
     65 063114  011001                9$:     MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
     66                                                               ;TABLE
     67 063116  001507                        BEQ     19$             ;BRANCH IF NO ERROR MESSAGE
     68
     69                                ;TYPE THE ERROR MESSAGE
     70 063120  012102                10$:    MOV     (R1)+,R2        ;R2=ADDRESS OF MESSAGE STRING
     71 063122  001505                        BEQ     19$             ;BRANCH IF END OF MESSAGE
     72 063124  010237  063272                MOV     R2,18$          ;LOAD ADDRESS OF STRING
     73 063130  005037  063464                CLR     BOTADR          ;CLEAR BOT ADDRESS
     74 063134  112203                11$:    MOVB    (R2)+,R3        ;END OF STRING??
     75 063136  001454                        BEQ     17$             ;YES!!
     76 063140  122703  000015                CMPB    #CR,R3          ;CARRIAGE RETURN??
     77 063144  001003                        BNE     12$             ;NO!!
     78 063146  105037  063467                CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
     79 063152  000770                        BR      11$             ;GET NEXT CHARACTER
     80 063154  122703  000012                CMPB    #LF,R3          ;LINE FEED??
     81 063160  001765                        BEQ     11$             ;YES-GET NEXT CHARACTER
     82 063162  122703  000011                CMPB    #HT,R3          ;HORIZONTAL TAB??
     83 063166  001007                        BNE     14$             ;NO!!
     84 063170  105237  063467                INCB    CHRCNT          ;ADJUST CHARACTER COUNT
     85 063174  132737  000007  063467        BITB    #7,CHRCNT
     86 063202  001372                        BNE     13$
     87 063204  000407                        BR      15$
     88 063206  105237  063467                INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
     89 063212  122703  000040                CMPB    #' ,R3          ;SPACE??
     90 063216  001002                        BNE     15$             ;NO!!
     91 063220  010237  063464                MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
     92 063224  122737  000100  063467        CMPB    #64.,CHRCNT     ;END OF LINE??
     93 063232  103340                        BHIS    11$             ;NO!!
     94 063234  013704  063464                MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
     95 063240  001007                        BNE     16$             ;BRANCH IF SPACE DETECTED
     96 063242  104401  001217                TYPE    ,$CRLF          ;TYPE CRLF
     97 063246  105037  063467                CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
     98 063252  013702  063272                MOV     18$,R2          ;SET UP R2 FOR TESTING
```

ERROR TYPEOUT ROUTINE

```
 99 063256  000726                          BR      11$
100 063260  105044                  16$:    CLRB    -(R4)           ;REPLACE SPACE
101 063262  112737  177777  063466          MOVB    #-1,BOTFLG      ;SET BOT FLAG
102 063270  104401                  17$:    TYPE                    ;TYPE ERROR MESSAGE STRING
103 063272  000000                  18$:    .WORD                   ;STRING ADDRESS GOES HERE
104 063274  105737  063466                  TSTB    BOTFLG          ;WAS STRING TRUNCATED??
105 063300  001707                          BEQ     10$             ;NO!!
106 063302  104401  001217                  TYPE    ,$CRLF          ;YES-TYPE CRLF
107 063306  105037  063466                  CLRB    BOTFLG          ;CLEAR BOT FLAG
108 063312  105037  063467                  CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
109 063316  013702  063464                  MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
110 063322  010237  063272                  MOV     R2,18$          ;SETUP 18$ FOR TYPING
111 063326  112742  000040                  MOVB    #' ,-(R2)       ;RESTORE SPACE
112 063332  105722                          TSTB    (R2)+           ;RESTORE R2
113 063334  000677                          BR      11$             ;TYPE REST OF STRING
114
115                                 ;TYPE ERROR HEADER AND ERROR DATA
116 063336  016001  000002          19$:    MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
117 063342  001444                          BEQ     27$             ;BRANCH IF NO HEADER
118 063344  104401  001217                  TYPE    ,$CRLF          ;(ASSUME NO DATA)
119 063350  016002  000004                  MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
120 063354  016003  000006                  MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
121 063360  012137  063370          20$:    MOV     (R1)+,21$       ;PUT HEADER ADDRESS FOR TYPE
122 063364  001433                          BEQ     27$             ;BRANCH IF END OF HEADERS
123                                                                 ;(ASSUME END OF DATA)
124 063366  104401                          TYPE
125 063370  000000                  21$:    .WORD   0               ;HEADER ADDRESS GOES HERE
126 063372  104401  001217                  TYPE    ,$CRLF
127 063376  005702                          TST     R2              ;DATA WITH HEADER??
128 063400  001767                          BEQ     20$             ;NO!!
129 063402  012204                          MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
130 063404  012305                          MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
131 063406  105725                  22$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
132 063410  100407                          BMI     24$             ;BINARY
133 063412  001403                          BEQ     23$             ;OCTAL
134 063414  013446                          MOV     @(R4)+,-(SP)    ;DECIMAL
135 063416  104405                          TYPDS
136 063420  000405                          BR      25$
137 063422  013446                  23$:    MOV     @(R4)+,-(SP)
138 063424  104402                          TYPOC
139 063426  000402                          BR      25$
140 063430  013446                  24$:    MOV     @(R4)+,-(SP)
141 063432  104406                          TYPBN
142 063434  005714                  25$:    TST     (R4)            ;MORE DATA??
143 063436  001403                          BEQ     26$             ;NO!!
144 063440  104401  063522                  TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
145 063444  000760                          BR      22$             ;AND CONTINUE
146 063446  104401  001217          26$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
147 063452  000742                          BR      20$             ;BEFORE NEXT HEADER
148 063454  104415                  27$:    RESREG
149 063456  000207                          RTS     PC
150
151 063460  000000                  TSTNMB: .WORD   0               ;TEST NUMBER
152 063462  000000                  ERRNMB: .WORD   0               ;ERROR NUMBER
153 063464  000000                  BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
154 063466     000                  BOTFLG: .BYTE   0               ;BOT FLAG
155 063467     000                  CHRCNT: .BYTE   0               ;CHARACTER COUNT
```

ERROR TYPEOUT ROUTINE

```
    156
    157 063470      104      122      126  ERTY00: .ASCIZ  @DRV#@
    158 063475      054      040      124  ERTY01: .ASCIZ  @, TEST#@
    159 063505      054      040      105  ERTY02: .ASCIZ  @, ERR#@
    160 063514      054      040      120  ERTY03: .ASCIZ  @, PC=@
    161 063522      040      040      000  ERTY04: .ASCIZ  @ @
    162 063525      040      055      040  ERTY05: .ASCIZ  @ - @
    163                                    .EVEN
    164 063532  063542  063630  063646  PFECH:  PFECH1,PFECH2,PFECH3,PFECH4 ;WORDS DEFINING TABLES BELOW
    165 063542  063546  000000          PFECH1: .+4,0
    166 063546      120      117      127          .ASCIZ  ?POWER MONITOR BIT IN CPU ERROR REGISTER FOUND SET?
    167                                          .EVEN
    168 063630  063634  000000          PFECH2: .+4,0
    169 063634      103      120      125          .ASCIZ  ?CPUERREG?
    170                                          .EVEN
    171 063646  063650                  PFECH3: .+2
    172 063650  062640  000000                   .WORD   CPSAVE,0
    173 063654  063656                  PFECH4: .+2
    174 063656      000      000                  .BYTE   0,0
```

```
     1                                  .SBTTL  TTY INPUT ROUTINE

                                        ;;*****************************************************************
                                        .ENABL  LSB
063660  000000                          $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
063662  000000                          $TKQIN: .WORD   0               ;;INPUT POINTER
063664  000000                          $TKQOUT: .WORD  0               ;;OUTPUT POINTER
063666                                  $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        063667                          $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
                                        ;
063670  005037  063660                  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
063674  012737  063666  063662                  MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
063702  013737  063662  063664                  MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
063710  012737  063740  000060                  MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
063716  012737  000200  000062                  MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
063724  005777  115232                          TST     @$TKB           ;;CLEAR DONE FLAG
063730  012777  000100  115222                  MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
063736  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)
                                        ;
063740  117746  115216                  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
063744  042716  177600                          BIC     #^C177,(SP)     ;;STRIP THE JUNK
063750  021627  000021                          CMP     (SP),#$XON      ;;IS IT A RANDOM XON?
063754  001002                                  BNE     30$             ;;BRANCH IF NO
063756  005726                                  TST     (SP)+           ;;CLEAN RANDOM XON OFF STACK
063760  000002                                  RTI                     ;;RETURN
063762                                  30$:
063762  021627  000003                          CMP     (SP),#3         ;;IS IT A CONTROL C?
063766  001007                                  BNE     1$              ;;BRANCH IF NO
063770  104401  065066                          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
063774  004737  063670                          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
064000  005726                                  TST     (SP)+           ;;CLEAN UP STACK
064002  000137  065130                          JMP     SHUT2           ;;CONTROL C RESTART
064006  021627  000007                  1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
064012  001004                                  BNE     2$              ;;BRANCH IF NO
064014  022737  000176  001154                  CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
064022  001500                                  BEQ     6$              ;;GO TO SWR CHANGE

064024                                  2$:
064024  022737  000001  063660                  CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
064032  001004                                  BNE     3$              ;;BRANCH IF NO
064034  104401  001212                          TYPE    ,$BELL          ;;RING THE TTY BELL
```

TTY INPUT ROUTINE

```
064040  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
064042  000451                          BR      5$              ;;EXIT
064044  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
064050  001021                          BNE     32$             ;;BRANCH IF NO
064052  005077  115102                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
064056  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
064060  105777  115074          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
064064  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
064066  117746  115070                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
064072  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
064076  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
064102  001366                          BNE     31$             ;;BRANCH IF NO
064104  012777  000100  115046          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
064112  000002                          RTI                     ;;RETURN
064114  005237  063660          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
064120  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
064124  002405                          BLT     4$              ;;BRANCH IF YES
064126  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
064132  003002                          BGT     4$              ;;BRANCH IF YES
064134  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
064140  112677  177516          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
064144  005237  063662                  INC     $TKQIN          ;;UPDATE THE POINTER
064150  023727  063662  063667          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
064156  001003                          BNE     5$              ;;BRANCH IF NO
064160  012737  063666  063662          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
064166  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.
064170  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
064176  001124                          BNE     15$             ;;EXIT IF NOT
064200  105777  114754                  TSTB    @$TKS           ;;IS A CHAR WAITING?
064204  100121                          BPL     15$             ;;IF NOT, EXIT
064206  117746  114750                  MOVB    @$TKB,-(SP)     ;;YES
064212  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
064216  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
064222  001300                          BNE     2$              ;;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.
064224  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
064232  001674                          BEQ     2$              ;;BRANCH IF YES
064234  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
064236  004737  063670                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
064242  005077  114712                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
064246  112737  000001  001151          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

064254  104401  065100                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
064260  104401  065105  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
064264  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
064270  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
064272  104401  065116                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
064276  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
064300  005046                          CLR     -(SP)           ;;THE NEW SWR
064302  105777  114652          7$:     TSTB    @$TKS           ;;CHAR THERE?
064306  100375                          BPL     7$              ;;IF NOT TRY AGAIN

064310  117746  114646                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
064314  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

064320  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
064324  001015                          BNE     9$              ;;BRANCH IF NOT
064326  104401  065066                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
064332  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
064336  123727  001151  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
064344  001003                          BNE     8$              ;;BRANCH IF NO
064346  012777  000100  114604          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
064354  000137  065130          8$:     JMP     SHUT2           ;;CONTROL-C RESTART


064360  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
064364  001005                          BNE     10$             ;;BRANCH IF NOT
064366  104401  065073                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
064372  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
064376  000737                          BR      19$             ;;LET'S TRY IT AGAIN


064400  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
064404  001022                          BNE     16$             ;;BRANCH IF NO
064406  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
064412  001403                          BEQ     11$             ;;BRANCH IF YES
064414  016677  000002  114532          MOV     2(SP),@SWR      ;;SAVE NEW SWR
064422  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
064426  104401  001217          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
064432  123727  001151  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
064440  001003                          BNE     15$             ;;BRANCH IF NOT
064442  012777  000100  114510          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
064450  000002                  15$:    RTI                     ;;RETURN
064452  004737  061346          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
064456  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
064462  002420                          BLT     18$             ;;BRANCH IF YES
064464  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
064470  003015                          BGT     18$             ;;BRANCH IF YES
064472  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
064476  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
064502  001403                          BEQ     17$             ;;BRANCH IF YES
064504  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
064506  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
064510  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
064512  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
064516  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
064522  000667                          BR      7$              ;;GET THE NEXT ONE
064524  104401  001216          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
064530  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
                                ;
064532  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
064534  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
064542  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
064546  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
064550  012746  064556                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
064554  000002                          RTI                     ;;POP NEW PC AND PS
064556                          64$:
064556  005737  063660          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
064562  001775                          BEQ     1$
064564  005337  063660                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
064570  117766  177070  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
064576  005237  063664                  INC     $TKQOUT         ;;UPDATE THE POINTER
064602  023727  063664  063667          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
064610  001003                          BNE     2$              ;;BRANCH IF NO
064612  012737  063666  063664          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
064620  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

064622  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
064624  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
064626  012703  065056          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
064632  022703  065066          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
064636  101456                          BLOS    4$              ;;BR IF YES
064640  104411                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
064642  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
064644  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
064650  001022                          BNE     5$              ;;BR IF NO
064652  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
064654  001007                          BNE     6$              ;;BR IF NO
064656  112737  000134  065054          MOVB    #'\,9$          ;;TYPE A BACK SLASH
064664  104401  065054                  TYPE    ,9$
064670  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
064674  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
064676  020327  065056                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
064702  103434                          BLO     4$              ;;BR IF YES
064704  111337  065054                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
064710  104401  065054                  TYPE    ,9$             ;;GO TYPE
064714  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
064716  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
064720  001406                          BEQ     7$              ;;BR IF NO
064722  112737  000134  065054          MOVB    #'\,9$          ;;TYPE A BACK SLASH
064730  104401  065054                  TYPE    ,9$
064734  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
064736  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
064742  001003                          BNE     8$              ;;BR IF NO
```

```
064744  104401  065073                  TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U''
064750  000726                          BR      1$              ;;GO START OVER
064752  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A ''^R''?
064756  001011                          BNE     3$              ;;BRANCH IF NO
064760  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
064762  104401  001217                  TYPE    ,$CRLF          ;;TYPE A ''CR'' & ''LF''
064766  104401  065056                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
064772  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
064774  104401  001216          4$:     TYPE    ,$QUES          ;;TYPE A '?'
065000  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
065002  111337  065054          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
065006  104401  065054                  TYPE    ,9$
065012  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
065016  001305                          BNE     2$              ;;LOOP IF NOT RETURN
065020  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
065024  104401  001220                  TYPE    ,$LF            ;;TYPE A LINE FEED
065030  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
065032  012603                          MOV     (SP)+,R3        ;;RESTORE R3
065034  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
065036  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
065044  012766  065056  000004          MOV     #$TTYIN,4(SP)
065052  000002                          RTI                     ;;RETURN
065054     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
065055     000                  .BYTE   0                       ;;TERMINATOR
065056                  $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
065066     136     103     015  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL ''C''
065073     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL ''U''
065100     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL ''G''
065105     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
065116     040     040     116  $MNEW:  .ASCIZ  /  NEW = /
                                .EVEN
2
3 065130  012737  177777  001326  SHUT2:  MOV     #-1,CTLFG       ;SET THE CONTROL-C FLAG
4 065136  105737  001116                  TSTB    $TSTNM          ;DOING ANY TESTS ?
5 065142  001002                          BNE     1$              ;BR IF YES
6 065144  000137  033776                  JMP     SHUT            ;NO--RESPOND TO ^C
7 065150  000002                  1$:     RTI                     ;EXIT FROM INTERRUPT
```

```
     1                                  .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

065152  011646                  $RDOCT: MOV     (SP),-(SP)          ;;PROVIDE SPACE FOR THE
065154  016666  000004  000002          MOV     4(SP),2(SP)         ;;INPUT NUMBER
065162  010046                          MOV     R0,-(SP)            ;;PUSH R0 ON STACK
065164  010146                          MOV     R1,-(SP)            ;;PUSH R1 ON STACK
065166  010246                          MOV     R2,-(SP)            ;;PUSH R2 ON STACK
065170  104412                  1$:     RDLIN                       ;;READ AN ASCIZ LINE
065172  012600                          MOV     (SP)+,R0            ;;GET ADDRESS OF 1ST CHARACTER
065174  005001                          CLR     R1                  ;;CLEAR DATA WORD
065176  005002                          CLR     R2
065200  112046                  2$:     MOVB    (R0)+,-(SP)         ;;PICKUP THIS CHARACTER
065202  001412                          BEQ     3$                  ;;IF ZERO GET OUT
065204  006301                          ASL     R1                  ;;*2
065206  006102                          ROL     R2
065210  006301                          ASL     R1                  ;;*4
065212  006102                          ROL     R2
065214  006301                          ASL     R1                  ;;*8
065216  006102                          ROL     R2
065220  042716  177770                  BIC     #^C7,(SP)           ;;STRIP THE ASCII JUNK
065224  062601                          ADD     (SP)+,R1            ;;ADD IN THIS DIGIT
065226  000764                          BR      2$                  ;;LOOP
065230  005726                  3$:     TST     (SP)+               ;;CLEAN TERMINATOR FROM STACK
065232  010166  000012                  MOV     R1,12(SP)           ;;SAVE THE RESULT
065236  010237  065252                  MOV     R2,$HIOCT
065242  012602                          MOV     (SP)+,R2            ;;POP STACK INTO R2
065244  012601                          MOV     (SP)+,R1            ;;POP STACK INTO R1
065246  012600                          MOV     (SP)+,R0            ;;POP STACK INTO R0
065250  000002                          RTI                         ;;RETURN
065252  000000              $HIOCT:     .WORD   0                   ;;HIGH ORDER BITS GO HERE
```

```
                                .SBTTL  TRAP DECODER

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

065254  016646  000002          $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
065260  042716  000020                  BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
065264  012746  065272                  MOV     #1$,-(SP)       ;; T-BIT TRAPS
065270  000002                          RTI                     ;;SET THE NEW STATUS
065272  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
065274  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
065300  005740                          TST     -(R0)           ;;BACKUP BY 2
065302  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
065304  006300                          ASL     R0              ;;POSITION FOR INDEXING
065306  016000  065326                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
065312  000200                          RTS     R0              ;;GO TO ROUTINE


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

065314  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
065316  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
065324  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
065326  065314                  $TRPAD: .WORD   $TRAP2
065330  061134                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
065332  060732                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
065334  060706                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
065336  060746                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
065340  060462                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
065342  060406                          $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

065344  064260                          $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING

065346  064170                          $CKSWR  ;;CALL=CKSWR    TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
065350  064532                          $RDCHR  ;;CALL=RDCHR    TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
065352  064622                          $RDLIN  ;;CALL=RDLIN    TRAP+12(104412) TTY TYPEIN STRING ROUTINE
065354  065152                          $RDOCT  ;;CALL=RDOCT    TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
065356  060312                          $SAVREG ;;CALL=SAVREG   TRAP+14(104414) SAVE R0-R5 ROUTINE
065360  060350                          $RESREG ;;CALL=RESREG   TRAP+15(104415) RESTORE R0-R5 ROUTINE
```

```
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
065362  012737  065522  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
065370  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
065376  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
065400  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
065402  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
065404  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
065406  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
065410  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
065412  017746  113536                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
065416  010637  065526                MOV     SP,$SAVR6       ;;SAVE SP
065422  012737  065434  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
065430  000000                        HALT
065432  000776                        BR      .-2             ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
065434  012737  065522  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
065442  013706  065526                MOV     $SAVR6,SP       ;;GET SP
065446  005037  065526                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
065452  005237  065526          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
065456  001375                        BNE     1$              ;;OF  WORD
065460  012677  113470                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
065464  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
065466  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
065470  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
065472  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
065474  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
065476  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
065500  012737  065362  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
065506  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
065514  104401                        TYPE                    ;REPORT THE POWER FAILURE
065516  065530          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
065520  000002                        RTI
065522  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
065524  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
065526  000000          $SAVR6: 0                       ;;PUT THE SP HERE
065530     015     012     120  $POWER: .ASCIZ  <15><12>'POWER'
                                .EVEN
```

```
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;*****************************************************************
065540  112737  000001  066004  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
065546  112737  000001  066002  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
065554  000403                  $ATYC   BR      $ATYC
065556  112737  000001  066004  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
065564                          $ATYC:
065564  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
065566  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
065570  105737  066002          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
065574  001400                  BEQ     5$              ;;IF NOT:  BR
065576  122737  000001  001242  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
065604  001031                  BNE     3$              ;;IF NOT:  BR
065606  132737  000100  001243  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
065614  001425                  BEQ     3$              ;;IF NOT:  BR
065616  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
065622  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
065630  005737  001222          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
065634  001375                  BNE     1$              ;;IF NOT:  WAIT
065636  010037  001236          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
065642  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
065644  001376                  BNE     2$
065646  163700  001236          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
065652  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
065654  010037  001240          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
065660  012737  000004  001222  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
065666  000413                  BR      5$
065670  017637  000004  065714  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
065676  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
065704  013746  177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
065710  004737  061134          JSR     PC,$TYPE        ;;CALL TYPE MACRO
065714  000000                  4$:     .WORD   0
065716                          5$:
065716  105737  066004          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
065722  001416                  BEQ     12$             ;;IF NOT:  BR
065724  005737  001242          TST     $ENV            ;;RUNNING UNDER APT?
065730  001413                  BEQ     12$             ;;IF NOT:  BR
065732  005737  001222          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
065736  001375                  BNE     11$             ;;IF NOT:  WAIT
065740  017637  000004  001224  MOV     @4(SP),$FATAL   ;;GET ERROR #
065746  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
065754  005237  001222          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
065760  105037  066004          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
065764  105037  066003          CLRB    $LFLG           ;;CLEAR LOG FLAG
065770  105037  066002          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
065774  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
065776  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
066000  000207                  RTS     PC              ;;RETURN
066002     000                  $MFLG:  .BYTE   0       ;;MESSG. FLAG
066003     000                  $LFLG:  .BYTE   0       ;;LOG FLAG
066004     000                  $FFLG:  .BYTE   0       ;;FATAL FLAG
                                        .EVEN
        000200                  APTSIZE = 200
        000001                  APTENV  = 001
        000100                  APTSPOOL= 100
        000040                  APTCSUP = 040
```

```
                                        .SBTTL  CONSOLE MESSAGES

     2
     3 066006  200  103  101  SCTMSG: .ASCIZ  <CRLF>@CANNOT RECOVER THE BAD SECTOR FILES ON THIS DEVICE@
     4 066072  075  000       EQUALS: .ASCIZ  @=@
     5 066074  101  114  114  ALL:    .ASCIZ  @ALL@<CRLF>
     6 066101  040  077  040  QUES:   .ASCIZ  @ ? @
     7 066105  054  040  000  COMMA:  .ASCIZ  @, @
     8 066110  200  124  117  MSHELP: .ASCII  <CRLF>@TO  ENSURE THAT  NO BAD HEADERS ARE LEFT  ON THE DISK@
     9 066176  200  104  125           .ASCII  <CRLF>@DURING  ERROR FREE OPERATION,  THIS PROGRAM SHOULD BE@
    10 066264  200  102  105           .ASCII  <CRLF>@BE  HALTED BY TYPING  A (^C) CONTROL C.  AS A RESULT,@
    11 066352  200  124  110           .ASCII  <CRLF>@THE PROGRAM WILL  BE HALTED WHEN THE DRIVE UNDER TEST@
    12 066440  200  110  101           .ASCII  <CRLF>@HAS COMPLETED TESTING.@
    13 066467  200                    .ASCII  <CRLF>
    14 066470  200  124  131           .ASCIZ  <CRLF>@TYPE HELP TEXT (L) N ? @
    15 066521  200  122  115  CNSL01: .ASCIZ  <CRLF>@RMCS1=@
    16 066531  040  114  111  CNSL02: .ASCIZ  @ LIMITS - LO= 160000, HI= 17XXXX@<CRLF>
    17 066573  122  115  126  CNSL03: .ASCIZ  @RMVEC=@
    18 066602  040  114  111  CNSL04: .ASCIZ  @ LIMITS - LO= 0, HI= 1000@<CRLF><LF>
    19 066636  200  124  131  CNSL07: .ASCII  <CRLF>@TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S)@
    20 066723  200  101  116           .ASCIZ  <CRLF>@AND TERMINATE INPUT WITH A CARRIAGE RETURN.@
    21 067000  200                    CNSL08: .ASCII  <CRLF>
    22 067001  040  077  111  CNSL09: .ASCIZ  @ ?ILLEGAL INPUT@<CRLF>
    23 067022  200  104  122  MSDRVS: .ASCIZ  <CRLF>/DRIVE(S): /
    24 067036  104  122  111  MSGDRV: .ASCIZ  /DRIVE/
    25 067044  200  125  116  SYSTAT: .ASCIZ  <CRLF>/UNIT STATUS:/
    26 067062  122  115  070  $RM80:  .ASCIZ  /RM80/
    27 067067  040  116  117  NOTRM:  .ASCIZ  / NOT AN RM80/
    28 067104  040  114  117  LODEV:  .ASCIZ  / LOAD DEVICE/
    29 067121  040  116  117  NOTPRS: .ASCIZ  / NOT PRESENT/
    30 067136  040  116  117  NOTAVL: .ASCIZ  / NOT AVAILABLE/
    31 067155  040  117  106  UNTOFF: .ASCIZ  / OFFLINE/
    32 067166  040  117  116  UNTON:  .ASCIZ  / ONLINE/
    33 067176  200  104  122  DRIVES: .ASCIZ  <CRLF>/DRIVE(S) TO BE TESTED/
    34 067225  116  117  116  NONE:   .ASCIZ  /NONE/
    35 067232  116  000       N:      .ASCIZ  /N/
    36 067234  131  000       Y:      .ASCIZ  /Y/
    37 067236  040            BLNKS4: .ASCII  / /
    38 067237  040            BLNKS3: .ASCII  / /
    39 06724C  040            BLNKS2: .ASCII  / /
    40 067241  040  000       BLNKS1: .ASCIZ  / /
    41                        .EVEN
```

```
                        .SBTTL  FUNCTION CODE TABLE

                        ;THE FUNCTION CODE TABLE IS USED TO DEFINE STATUS CONDITIONS FOR
                        ;EACH FUNCTION CODE.  BIT USAGE IS AS FOLLOWS:

                        ;       ATA - BIT 15 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF ATA SHOULD BE SET WHEN THE FUNCTION CODE IS EXECUTED, OTHERWISE,
                        ;BIT 15 IS ZERO, INDICATING THAT ATA SHOULD NOT NORMALLY BE SET.
                        ;NOTE THAT ATA MAY BE SET WHEN A COMMAND IS EXECUTED EVEN THOUGH
                        ;IT IS NOT EXPECTED AS A RESULT OF THE COMMAND.

                        ;       WCE - BIT 14 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF WRITE CHECK ERRORS ARE ENABLED AS A FUNCTION OF THE COMMAND.

                        ;       OPI - BIT 13 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF OPI ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.

                        ;       IVC - BIT 12 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF IVC ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.

                        ;       WLE - BIT 11 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF WRITE ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.
                        ;THE WRITE ERRORS WHICH ARE ENABLED ARE "WLE", "WCF", "DPE", "UPE".

                        ;       IAE - BIT 10 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF INVALID ADDRESS ERROR IS ENABLED FOR THAT COMMAND.

                        ;       AOE - BIT 09 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF READ AND WRITE ERRORS ARE ENABLED DURING THE EXECUTION OF THE
                        ;COMMAND.  THE ERRORS ENABLED BY THIS BIT ARE "TRE", "DLT", "NEM",
                        ;"MXF", "LBT", AND "AOE".

                        ;               BIT 08 IS NOT USED.

                        ;       HCE - BIT 07 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF HEADER ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.
                        ;HEADER ERRORS INCLUDE "HCRC", "HCE", "FER", AND "BSE".

                        ;       ECH - BIT 06 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                        ;IF DATA FIELD ERRORS ARE ENABLED DURING THE EXECUTION OF THAT
                        ;COMMAND.  THESE ERRORS INCLUDE "MDPE", "DCK", AND "ECH".

                        ;               BIT 05 IS NOT USED.

                        ;               BIT 04 IS NOT USED.

                        ;               BIT 03 IS NOT USED.

                        ;               BIT 02 IS NOT USED.

                        ;               BIT 01 IS NOT USED.

                        ;       ILF - BIT 00 IS SET IF THE FUNCTION CODE IS ILLEGAL.

067244                  FNCDTB:                                 ;FUNCTION CODE TABLE

067244  020000          .WORD   OPI                             ;NOP
```

FUNCTION CODE TABLE

```
     58 067246  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (2)
     59 067250  132000                .WORD   ATA!OPI!IVC!IAE                        ;SEEK
     60 067252  130000                .WORD   ATA!OPI!IVC                            ;RECALIBRATE
     61 067254  020000                .WORD   OPI                                    ;DRIVE CLEAR
     62 067256  030000                .WORD   OPI!IVC                                ;RELEASE
     63 067260  130000                .WORD   OPI!ATA!IVC                            ;OFFSET
     64 067262  130000                .WORD   OPI!ATA!IVC                            ;RETURN TO CENTERLINE
     65 067264  020000                .WORD   OPI                                    ;READ IN PRESET
     66 067266  020000                .WORD   OPI                                    ;PACK ACKNOWLEDGE
     67 067270  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (24)
     68 067272  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (26)
     69 067274  132000                .WORD   ATA!OPI!IVC!IAE                        ;SEARCH
     70 067276  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (32)
     71 067300  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (34)
     72 067302  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (36)
     73 067304  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (40)
     74 067306  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (42)
     75 067310  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (44)
     76 067312  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (46)
     77 067314  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH            ;WRITE CHECK DATA
     78 067316  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH            ;WRITE CHECK HEADER AND DATA
     79 067320  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (54)
     80 067322  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (56)
     81 067324  037200                .WORD   OPI!IVC!WLE!IAE!AOE!HCE                ;WRITE DATA
     82 067326  037000                .WORD   OPI!IVC!WLE!IAE!AOE                    ;WRITE HEADER AND DATA
     83 067330  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (64)
     84 067332  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (66)
     85 067334  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH                ;READ DATA
     86 067336  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH                ;READ HEADER AND DATA
     87 067340  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (74)
     88 067342  130001                .WORD   OPI!ATA!ILF!IVC                        ;ILLEGAL FUNCTION (76)
```

ATTENTION (ATA) TABLE

```
                                .SBTTL  ATTENTION (ATA) TABLE

                        ATNTBL: .BYTE   1.
                                .BYTE   2.
                                .BYTE   4.
                                .BYTE   8.
                                .BYTE   16.
                                .BYTE   32.
                                .BYTE   64.
                                .BYTE   128.
```

DATA PATTERN TABLE

```
                                .SBTTL  DATA PATTERN TABLE

                        RGDTPT:
                        MIXED:
067354 000000                   .WORD   0.
067356 000001                   .WORD   1.
067360 000003                   .WORD   3.
067362 000007                   .WORD   7.
067364 000017                   .WORD   15.
067366 000037                   .WORD   31.
067370 000077                   .WORD   63.
067372 000177                   .WORD   127.
067374 000377                   .WORD   255.
067376 000777                   .WORD   511.
067400 001777                   .WORD   1023.
067402 003777                   .WORD   2047.
067404 007777                   .WORD   4095.
067406 017777                   .WORD   8191.
067410 037777                   .WORD   16383.
067412 077777                   .WORD   32767.
067414 177777           ONES:   .WORD   65535.
067416 177777                   .WORD   65535.
067420 077777                   .WORD   32767.
067422 037777                   .WORD   16383.
067424 017777                   .WORD   8191.
067426 007777                   .WORD   4095.
067430 003777                   .WORD   2047.
067432 001777                   .WORD   1023.
067434 000777                   .WORD   511.
067436 000377                   .WORD   255.
067440 000177                   .WORD   127.
067442 000077                   .WORD   63.
067444 000037                   .WORD   31.
067446 000017                   .WORD   15.
067450 000007                   .WORD   7.
067452 000003                   .WORD   3.
067454 000001                   .WORD   1.
067456 000000           ZEROS:  .WORD   0.
067460 000000                   .WORD   0.
067462 000001                   .WORD   1.
067464 000002                   .WORD   2.
067466 000004                   .WORD   4.
067470 000010                   .WORD   8.
067472 000020                   .WORD   16.
067474 000040                   .WORD   32.
067476 000100                   .WORD   64.
067500 000200                   .WORD   128.
067502 000400                   .WORD   256.
067504 001000                   .WORD   512.
067506 002000                   .WORD   1024.
067510 004000                   .WORD   2048.
067512 010000                   .WORD   4096.
067514 020000                   .WORD   8192.
067516 040000                   .WORD   16384.
067520 100000                   .WORD   32768.
067522 100000                   .WORD   32768.
067524 040000                   .WORD   16384.
```

DATA PATTERN TABLE

```
     58 067526  020000                          .WORD   8192.
     59 067530  010000                          .WORD   4096.
     60 067532  004000                          .WORD   2048.
     61 067534  002000                          .WORD   1024.
     62 067536  001000                          .WORD   512.
     63 067540  000400                          .WORD   256.
     64 067542  000200                          .WORD   128.
     65 067544  000100                          .WORD   64.
     66 067546  000040                          .WORD   32.
     67 067550  000020                          .WORD   16.
     68 067552  000010                          .WORD   8.
     69 067554  000004                          .WORD   4.
     70 067556  000002                          .WORD   2.
     71 067560  000001                          .WORD   1.
     72 067562  000000                          .WORD   0.
     73 067564  177777                          .WORD   65535.
     74 067566  177776                          .WORD   65534.
     75 067570  177774                          .WORD   65532.
     76 067572  177770                          .WORD   65528.
     77 067574  177760                          .WORD   65520.
     78 067576  177740                          .WORD   65504.
     79 067600  177700                          .WORD   65472.
     80 067602  177600                          .WORD   65408.
     81 067604  177400                          .WORD   65280.
     82 067606  177000                          .WORD   65024.
     83 067610  176000                          .WORD   64512.
     84 067612  174000                          .WORD   63488.
     85 067614  170000                          .WORD   61440.
     86 067616  160000                          .WORD   57344.
     87 067620  140000                          .WORD   49152.
     88 067622  100000                          .WORD   32768.
     89 067624  000000                          .WORD   0.
     90 067626  000000                          .WORD   0.
     91 067630  100000                          .WORD   32768.
     92 067632  140000                          .WORD   49152.
     93 067634  160000                          .WORD   57344.
     94 067636  170000                          .WORD   61440.
     95 067640  174000                          .WORD   63488.
     96 067642  176000                          .WORD   64512.
     97 067644  177000                          .WORD   65024.
     98 067646  177400                          .WORD   65280.
     99 067650  177600                          .WORD   65408.
    100 067652  177700                          .WORD   65472.
    101 067654  177740                          .WORD   65504.
    102 067656  177760                          .WORD   65520.
    103 067660  177770                          .WORD   65528.
    104 067662  177774                          .WORD   65532.
    105 067664  177776                          .WORD   65534.
    106 067666  177777                          .WORD   65535.
    107 067670  125252                EARLY:    .WORD   43690.
    108 067672  152525                          .WORD   43690./2
    109 067674  125252                          .WORD   43690.
    110 067676  177777                          .WORD   65535.
    111 067700  177776                          .WORD   65534.
    112 067702  177775                          .WORD   65533.
    113 067704  177773                          .WORD   65531.
    114 067706  177767                          .WORD   65527.
```

DATA PATTERN TABLE

```
    115 067710  177757                          .WORD   65519.
    116 067712  177737                          .WORD   65503.
    117 067714  177677                          .WORD   65471.
    118 067716  177577                          .WORD   65407.
    119 067720  177377                          .WORD   65279.
    120 067722  176777                          .WORD   65023.
    121 067724  175777                          .WORD   64511.
    122 067726  173777                          .WORD   63487.
    123 067730  167777                          .WORD   61439.
    124 067732  157777                          .WORD   57343.
    125 067734  137777                          .WORD   49151.
    126 067736  077777                          .WORD   32767.
    127 067740  077777                          .WORD   32767.
    128 067742  137777                          .WORD   49151.
    129 067744  157777                          .WORD   57343.
    130 067746  167777                          .WORD   61439.
    131 067750  173777                          .WORD   63487.
    132 067752  175777                          .WORD   64511.
    133 067754  176777                          .WORD   65023.
    134 067756  177377                          .WORD   65279.
    135 067760  177577                          .WORD   65407.
    136 067762  177677                          .WORD   65471.
    137 067764  177737                          .WORD   65503.
    138 067766  177757                          .WORD   65519.
    139 067770  177767                          .WORD   65527.
    140 067772  177773                          .WORD   65531.
    141 067774  177775                          .WORD   65533.
    142 067776  177776                          .WORD   65534.
    143 070000  177777                          .WORD   65535.
    144 070002                          ENRGDT:
```

ERROR MESSAGE TABLE

```
                                        .SBTTL  ERROR MESSAGE TABLE
                                        
070002  074442  000000          EMT1:   .WORD   EMS1,0
070006  074511  074526  000000  EMT2:   .WORD   EMS2,EMS3,0
070014  074511  074571  000000  EMT3:   .WORD   EMS2,EMS4,0
070022  074634  074664  000000  EMT4:   .WORD   EMS5,EMS6,0
070030  074634  074776  000000  EMT5:   .WORD   EMS5,EMS10,0
070036  101427  076657  000000  EMT6:   .WORD   EMS167,EMS64,0
070044  077425  101454  000000  EMT7:   .WORD   EMS110,EMS170,0
070052  074731  000000          EMT10:  .WORD   EMS7,0
070056  074776  000000          EMT11:  .WORD   EMS10,0
070062  075040  075051  000000  EMT12:  .WORD   EMS11,EMS12,0
070070  075112  075123  075134  EMT13:  .WORD   EMS13,EMS14,EMS15,EMS16,0
070102  075206  076657  000000  EMT14:  .WORD   EMS17,EMS64,0
070110  075040  075271  000000  EMT15:  .WORD   EMS11,EMS21,0
070116  075040  075314  075443  EMT16:  .WORD   EMS11,EMS22,EMS27,0
070126  075040  075330  075454  EMT17:  .WORD   EMS11,EMS23,EMS30,0
070136  075040  075356  075454  EMT20:  .WORD   EMS11,EMS24,EMS30,0
070146  075040  075405  075443  EMT21:  .WORD   EMS11,EMS25,EMS27,0
070156  075040  075422  075443  EMT22:  .WORD   EMS11,EMS26,EMS27,0
070166  075040  075464  075454  EMT23:  .WORD   EMS11,EMS31,EMS30,0
070176  075040  075513  075454  EMT24:  .WORD   EMS11,EMS32,EMS30,0
070206  075040  075542  075454  EMT25:  .WORD   EMS11,EMS33,EMS30,0
070216  075040  075570  075454  EMT26:  .WORD   EMS11,EMS34,EMS30,0
070226  075040  075641  075454  EMT27:  .WORD   EMS11,EMS35,EMS30,0
070236  075040  075670  075454  EMT30:  .WORD   EMS11,EMS36,EMS30,0
070246  075040  075717  075454  EMT31:  .WORD   EMS11,EMS37,EMS30,0
070256  075040  075745  075454  EMT32:  .WORD   EMS11,EMS40,EMS30,0
070266  075040  075774  075454  EMT33:  .WORD   EMS11,EMS41,EMS30,0
070276  075040  076022  075454  EMT34:  .WORD   EMS11,EMS42,EMS30,0
070306  075040  076051  075454  EMT35:  .WORD   EMS11,EMS43,EMS30,0
070316  075040  076100  075454  EMT36:  .WORD   EMS11,EMS44,EMS30,0
070326  075040  076153  075454  EMT37:  .WORD   EMS11,EMS45,EMS30,0
070336  076743  075246  000000  EMT40:  .WORD   EMS66,EMS20,0
070344  077131  100631  077137  EMT41:  .WORD   EMS75,EMS141,EMS76,0
070354  100722  100732  077065  EMT42:  .WORD   EMS144,EMS145,EMS72,EMS76,0
070366  076245  076363  077137  EMT43:  .WORD   EMS47,EMS53,EMS76,0
070376  077170  076363  077137  EMT44:  .WORD   EMS77,EMS53,EMS76,0
070406  077216  076363  077137  EMT45:  .WORD   EMS100,EMS53,EMS76,0
070416  077244  076363  077137  EMT46:  .WORD   EMS101,EMS53,EMS76,0
070426  076675  076657  000000  EMT47:  .WORD   EMS65,EMS64,0
070434  075112  075134  076631  EMT50:  .WORD   EMS13,EMS15,EMS63,0
070444  075040  076216  075454  EMT51:  .WORD   EMS11,EMS46,EMS30,EMS67,0
070456  076245  076363  077013  EMT52:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS140,EMS141,0
070474  076245  076363  077013  EMT53:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS141,EMS164,0
070512  076274  076363  077013  EMT54:  .WORD   EMS50,EMS53,EMS67,0
070522  100657  100674  076363  EMT55:  .WORD   EMS142,EMS143,EMS53,EMS67,0
070534  076323  077065  077013  EMT56:  .WORD   EMS51,EMS72,EMS67,EMS115,EMS50,EMS70,0
070552  101427  077075  077013  EMT57:  .WORD   EMS167,EMS73,EMS67,0
070562  076446  077013  077625  EMT60:  .WORD   EMS56,EMS67,EMS115,EMS150,EMS152,EMS70,0
070600  077052  076446  077013  EMT61:  .WORD   EMS71,EMS56,EMS67,EMS115,EMS150,EMS152,EMS72,0
070620  100610  077013  077625  EMT62:  .WORD   EMS140,EMS67,EMS115,EMS47,EMS70,0
070634  102627  102652  075246  EMT63:  .WORD   EMS225,EMS226,EMS20,0
070644  101015  101057  077040  EMT64:  .WORD   EMS150,EMS152,EMS70,EMS115,EMS56,EMS73,EMS67,0
070664  076351  102054  102235  EMT65:  .WORD   EMS52,EMS205,EMS214,EMS206,EMS115,EMS51,EMS72,0
070704  101366  077321  077065  EMT66:  .WORD   EMS165,EMS103,EMS72,EMS124,0
070716  101366  077321  077065  EMT67:  .WORD   EMS165,EMS103,EMS72,EMS171,0
```

ERROR MESSAGE TABLE

```
 58 070730  076216  075246  101261  EMT70:   .WORD   EMS46,EMS20,EMS163,0
 59 070740  077052  077244  101261  EMT71:   .WORD   EMS71,EMS101,EMS163,0
 60 070750  076245  077065  101261  EMT72:   .WORD   EMS47,EMS72,EMS163,EMS115,EMS140,EMS141,0
 61 070766  076245  077065  101261  EMT73:   .WORD   EMS47,EMS72,EMS163,EMS115,EMS141,EMS72,0
 62 071004  076446  076363  101261  EMT74:   .WORD   EMS56,EMS53,EMS163,0
 63 071014  077052  076446  101261  EMT75:   .WORD   EMS71,EMS56,EMS163,EMS115,EMS150,EMS152,EMS72,0
 64 071034  076274  076363  101261  EMT76:   .WORD   EMS50,EMS53,EMS163,0
 65 071044  100657  100674  076363  EMT77:   .WORD   EMS142,EMS143,EMS53,EMS163,0
 66 071056  077131  100631  101261  EMT100:  .WORD   EMS75,EMS141,EMS163,EMS115,EMS47,EMS70,0
 67 071074  077131  101015  101261  EMT101:  .WORD   EMS75,EMS150,EMS163,EMS115,EMS56,EMS73,0
 68 071112  101427  077075  101261  EMT102:  .WORD   EMS167,EMS73,EMS163,0
 69 071122  101000  101033  077112  EMT103:  .WORD   EMS147,EMS151,EMS74,EMS163,0
 70 071134  077323  077112  101261  EMT104:  .WORD   EMS51,EMS74,EMS163,EMS115,EMS50,EMS70,0
 71 071146  100566  077216  101261  EMT105:  .WORD   EMS165,EMS100,EMS163,0
 72 071162  101366  077170  101261  EMT106:  .WORD   EMS165,EMS77,EMS163,0
 73 071172  100722  100732  076363  EMT107:  .WORD   EMS144,EMS145,EMS53,EMS163,EMS115,EMS143,EMS70,0
 74 071212  077425  077465  077630  EMT110:  .WORD   EMS110,EMS112,EMS116,EMS111,0
 75 071224  077551  074571  000000  EMT111:  .WORD   EMS113,EMS4,0
 76 071232  077551  074526  000000  EMT112:  .WORD   EMS113,EMS3,0
 77 071240  077425  077625  077630  EMT113:  .WORD   EMS110,EMS115,EMS116,EMS117,EMS114,0
 78 071254  000000                  EMT114:  .WORD
 79 071256  077702  100342  100366  EMT115:  .WORD   EMS121,EMS132,EMS133,0
 80 071266  077745  100342  100366  EMT116:  .WORD   EMS122,EMS132,EMS133,0
 81 071276  100002  100342  100366  EMT117:  .WORD   EMS123,EMS132,EMS133,0
 82 071306  100045  100342  100366  EMT120:  .WORD   EMS124,EMS132,EMS133,0
 83 071316  100077  100342  100366  EMT121:  .WORD   EMS125,EMS132,EMS133,0
 84 071326  100142  100342  100366  EMT122:  .WORD   EMS126,EMS132,EMS133,0
 85 071336  100553  100342  100366  EMT123:  .WORD   EMS137,EMS132,EMS133,0
 86 071346  100244  100342  100366  EMT124:  .WORD   EMS130,EMS132,EMS133,0
 87 071356  100302  100342  100366  EMT125:  .WORD   EMS131,EMS132,EMS133,0
 88 071366  077702  100342  100411  EMT126:  .WORD   EMS121,EMS132,EMS134,EMS123,0
 89 071400  077745  100342  100411  EMT127:  .WORD   EMS122,EMS132,EMS134,EMS123,0
 90 071412  100002  100342  100411  EMT130:  .WORD   EMS123,EMS132,EMS134,EMS123,0
 91 071424  100045  100342  100411  EMT131:  .WORD   EMS124,EMS132,EMS134,EMS123,0
 92 071436  100077  100342  100411  EMT132:  .WORD   EMS125,EMS132,EMS134,EMS123,0
 93 071450  100142  100342  100411  EMT133:  .WORD   EMS126,EMS132,EMS134,EMS123,0
 94 071462  100553  100342  100411  EMT134:  .WORD   EMS137,EMS132,EMS134,EMS123,0
 95 071474  100244  100342  100411  EMT135:  .WORD   EMS130,EMS132,EMS134,EMS123,0
 96 071506  100302  100342  100411  EMT136:  .WORD   EMS131,EMS132,EMS134,EMS123,0
 97 071520  077702  100342  100453  EMT137:  .WORD   EMS121,EMS132,EMS135,0
 98 071530  100002  100342  100453  EMT140:  .WORD   EMS123,EMS132,EMS135,0
 99 071540  077702  100342  100526  EMT141:  .WORD   EMS121,EMS132,EMS136,0
100 071550  100553  100342  100526  EMT142:  .WORD   EMS137,EMS132,EMS136,0
101 071560  100045  100342  100526  EMT143:  .WORD   EMS124,EMS132,EMS136,0
102 071570  100077  100342  100526  EMT144:  .WORD   EMS125,EMS132,EMS136,0
103 071600  100142  100342  100526  EMT145:  .WORD   EMS126,EMS132,EMS136,0
104 071610  100302  100342  100526  EMT146:  .WORD   EMS131,EMS132,EMS136,0
105 071620  101503  100342  100526  EMT147:  .WORD   EMS171,EMS132,EMS136,0
106 071630  100244  100342  100526  EMT150:  .WORD   EMS130,EMS132,EMS136,0
107 071640  100610  077625  100631  EMT151:  .WORD   EMS140,EMS115,EMS141,EMS70,0
108 071652  100657  077625  100674  EMT152:  .WORD   EMS142,EMS115,EMS143,EMS72,0
109 071664  100722  100732  100761  EMT153:  .WORD   EMS144,EMS145,EMS146,EMS115,EMS143,EMS72,0
110 071702  100722  100732  075246  EMT154:  .WORD   EMS144,EMS145,EMS20,EMS115,EMS143,EMS70,0
111 071720  101015  101057  101125  EMT155:  .WORD   EMS150,EMS152,EMS154,EMS153,0
112 071732  101000  101033  101125  EMT156:  .WORD   EMS147,EMS151,EMS154,EMS155,0
113 071744  101000  101033  101161  EMT157:  .WORD   EMS147,EMS151,EMS156,EMS157,0
114 071756  101231  101161  101214  EMT160:  .WORD   EMS161,EMS156,EMS160,0
```

EPROR MESSAGE TABLE

```
115 071766  075405  075443  101161  EMT161: .WORD   EMS25,EMS27,EMS156,EMS160,0
116 072000  075422  075443  101161  EMT162: .WORD   EMS26,EMS27,EMS156,EMS160,0
117 072012  076245  101261  100610  EMT163: .WORD   EMS47,EMS163,EMS140,0
118 072022  076245  075246  000000  EMT164: .WORD   EMS47,EMS20,0
119 072030  076446  075246  000000  EMT165: .WORD   EMS56,EMS20,0
120 072036  076216  075246  000000  EMT166: .WORD   EMS46,EMS20,0
121 072044  102555  075246  000000  EMT167: .WORD   EMS224,EMS20,0
122 072052  102013  100631  075454  EMT170: .WORD   EMS203,EMS141,EMS30,EMS202,0
123 072064  101427  101125  101177  EMT171: .WORD   EMS167,EMS154,EMS157,0
124 072074  101427  101125  101141  EMT172: .WORD   EMS167,EMS154,EMS155,0
125 072104  101503  100342  100366  EMT173: .WORD   EMS171,EMS132,EMS133,0
126 072114  101503  100342  100411  EMT174: .WORD   EMS171,EMS132,EMS134,EMS123,0
127 072126  077551  101656  000000  EMT175: .WORD   EMS113,EMS177,0
128 072134  101700  101715  000000  EMT176: .WORD   EMS200,EMS201,0
129 072142  000000                  EMT177: .WORD
130 072144  102013  076323  075454  EMT200: .WORD   EMS203,EMS51,EMS30,EMS202,0
131 072156  102013  102026  075454  EMT201: .WORD   EMS203,EMS204,EMS30,EMS202,0
132 072170  102013  076274  075454  EMT202: .WORD   EMS203,EMS50,EMS30,EMS202,0
133 072202  102013  100674  075454  EMT203: .WORD   EMS203,EMS143,EMS30,EMS202,0
134 072214  101015  077112  101763  EMT204: .WORD   EMS150,EMS74,EMS202,EMS115,EMS152,EMS72,0
135 072232  076274  077321  077065  EMT205: .WORD   EMS50,EMS103,EMS72,0
136 072242  077330  077075  101763  EMT206: .WORD   EMS104,EMS73,EMS202,0
137 072252  077131  100631  077625  EMT207: .WORD   EMS75,EMS141,EMS115,EMS140,0
138 072264  077131  101015  000000  EMT210: .WORD   EMS75,EMS150,0
139 072272  076323  077065  077625  EMT211: .WORD   EMS51,EMS72,EMS115,EMS50,EMS70,0
140 072306  100657  076363  077625  EMT212: .WORD   EMS142,EMS53,EMS115,EMS143,EMS72,0
141 072322  076274  077321  076363  EMT213: .WORD   EMS50,EMS103,EMS53,0
142 072332  076351  102054  101402  EMT214: .WORD   EMS52,EMS205,EMS166,EMS206,EMS115,EMS51,EMS72,0
143 072352  076351  077660  076446  EMT215: .WORD   EMS52,EMS117,EMS56,EMS163,0
144 072364  076446  075454  076657  EMT216: .WORD   EMS56,EMS30,EMS64,0
145 072374  076216  075454  076657  EMT217: .WORD   EMS46,EMS30,EMS64,0
146 072404  075464  075454  076657  EMT220: .WORD   EMS31,EMS30,EMS64,0
147 072414  076245  075454  076657  EMT221: .WORD   EMS47,EMS30,EMS64,0
148 072424  076351  077660  077170  EMT222: .WORD   EMS52,EMS117,EMS77,0
149 072434  076351  077660  076631  EMT223: .WORD   EMS52,EMS117,EMS63,0
150 072444  076351  077660  102327  EMT224: .WORD   EMS52,EMS117,EMS225,EMS115,EMS226,0
151 072460  102627  102701  102712  EMT225: .WORD   EMS225,EMS227,EMS230,EMS115,EMS156,EMS231,EMS57,0
152 072500  075040  102712  075454  EMT226: .WORD   EMS11,EMS230,EMS30,0
153 072510  102627  102652  077625  EMT227: .WORD   EMS225,EMS226,EMS115,EMS230,EMS72,0
154 072524  000000                  EMT230: .WORD
155 072526  000000                  EMT231: .WORD
156 072530  000000                  EMT232: .WORD
157 072532  000000                  EMT233: .WORD
158 072534  000000                  EMT234: .WORD
159 072536  000000                  EMT235: .WORD
160 072540  000000                  EMT236: .WORD
161 072542  000000                  EMT237: .WORD
162 072544  000000                  EMT240: .WORD
163 072546  000000                  EMT241: .WORD
164 072550  000000                  EMT242: .WORD
165 072552  000000                  EMT243: .WORD
166 072554  000000                  EMT244: .WORD
167 072556  000000                  EMT245: .WORD
168 072560  101427  100342  102113  EMT246: .WORD   EMS167,EMS132,EMS207,0
169 072570  101427  100342  102140  EMT247: .WORD   EMS167,EMS132,EMS210,EMS125,0
170 072602  101427  102151  102140  EMT250: .WORD   EMS167,EMS211,EMS210,EMS207,EMS206,0
171 072616  102167  102212  000000  EMT251: .WORD   EMS212,EMS213,0
```

ERROR MESSAGE TABLE

```
172 072624  101366  102167  000000  EMT252: .WORD   EMS165,EMS212,0
173 072632  101000  101033  101161  EMT253: .WORD   EMS147,EMS151,EMS156,EMS210,EMS26,EMS27,0
174 072650  102013  077244  075454  EMT254: .WORD   EMS203,EMS101,EMS30,0
175 072660  102013  101427  075454  EMT255: .WORD   EMS203,EMS167,EMS30,0
176 072670  102013  077216  075454  EMT256: .WORD   EMS203,EMS100,EMS30,0
177 072700  075040  076216  075454  EMT257: .WORD   EMS11,EMS46,EMS30,EMS102,0
178 072712  076351  077660  076446  EMT260: .WORD   EMS52,EMS117,EMS56,EMS102,0
179 072724  076351  102054  102357  EMT261: .WORD   EMS52,EMS205,EMS220,EMS206,EMS115,EMS51,EMS72,0
180 072744  077330  077075  101763  EMT262: .WORD   EMS104,EMS73,EMS202,0
181 072754  076274  076363  077272  EMT263: .WORD   EMS50,EMS53,EMS102,0
182 072764  076446  076363  077272  EMT264: .WORD   EMS56,EMS53,EMS102,EMS115,EMS150,EMS152,EMS70,0
183 073004  077052  076446  077272  EMT265: .WORD   EMS71,EMS56,EMS102,EMS115,EMS150,EMS152,EMS72,0
184 073024  100657  100674  076363  EMT266: .WORD   EMS142,EMS143,EMS53,EMS102,0
185 073036  076274  100761  077625  EMT267: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,0
186 073054  076245  076363  077272  EMT270: .WORD   EMS47,EMS53,EMS102,EMS115,EMS140,0
187 073070  076245  076363  077272  EMT271: .WORD   EMS47,EMS53,EMS102,EMS115,EMS141,EMS72,0
188 073106  077131  100631  077272  EMT272: .WORD   EMS75,EMS141,EMS102,EMS115,EMS47,EMS73,0
189 073124  076323  077112  077272  EMT273: .WORD   EMS51,EMS74,EMS102,EMS115,EMS50,EMS70,0
190 073142  076631  076363  076523  EMT274: .WORD   EMS63,EMS53,EMS57,EMS115,EMS41,EMS146,0
191 073160  077630  076363  075774  EMT275: .WORD   EMS116,EMS53,EMS41,EMS57,0
192 073172  076245  076363  076523  EMT276: .WORD   EMS47,EMS53,EMS57,EMS115,EMS140,0
193 073206  076245  076363  076523  EMT277: .WORD   EMS47,EMS53,EMS57,EMS115,EMS141,EMS72,0
194 073224  076446  076363  076523  EMT300: .WORD   EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS70,0
195 073244  077052  076446  076363  EMT301: .WORD   EMS71,EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS72,0
196 073266  101366  077216  077321  EMT302: .WORD   EMS165,EMS100,EMS103,EMS57,0
197 073300  101366  077244  077321  EMT303: .WORD   EMS165,EMS101,EMS103,EMS57,0
198 073312  101366  077170  077321  EMT304: .WORD   EMS165,EMS77,EMS103,EMS57,0
199 073324  076216  075454  076657  EMT305: .WORD   EMS46,EMS30,EMS64,EMS57,0
200 073336  076274  076363  076523  EMT306: .WORD   EMS50,EMS53,EMS57,0
201 073346  076274  100761  077625  EMT307: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,EMS57,0
202 073366  100657  100674  076363  EMT310: .WORD   EMS142,EMS143,EMS53,EMS57,0
203 073400  077330  077321  076363  EMT311: .WORD   EMS104,EMS103,EMS53,EMS57,0
204 073412  077357  077321  076363  EMT312: .WORD   EMS105,EMS103,EMS53,EMS57,0
205 073424  100722  100732  077321  EMT313: .WORD   EMS144,EMS145,EMS103,EMS57,EMS115,EMS143,EMS70,0
206 073444  076022  077321  076363  EMT314: .WORD   EMS42,EMS103,EMS53,EMS57,0
207 073456  075464  077321  076363  EMT315: .WORD   EMS31,EMS103,EMS53,EMS57,0
208 073470  077052  075464  077321  EMT316: .WORD   EMS71,EMS31,EMS103,EMS57,0
209 073502  076051  077321  076523  EMT317: .WORD   EMS43,EMS103,EMS57,0
210 073512  076153  077321  076523  EMT320: .WORD   EMS45,EMS103,EMS57,0
211 073522  076100  077321  076523  EMT321: .WORD   EMS44,EMS103,EMS57,0
212 073532  077406  075246  000000  EMT322: .WORD   EMS106,EMS20,0
213 073540  075670  077321  076523  EMT323: .WORD   EMS36,EMS103,EMS57,0
214 073550  101556  075670  077321  EMT324: .WORD   EMS173,EMS36,EMS103,EMS57,0
215 073562  101536  075670  077321  EMT325: .WORD   EMS172,EMS36,EMS103,EMS57,0
216 073574  075112  101573  075134  EMT326: .WORD   EMS13,EMS174,EMS15,EMS35,EMS53,EMS175,0
217 073612  101000  101033  077112  EMT327: .WORD   EMS147,EMS151,EMS74,EMS175,0
218 073624  076743  076363  101601  EMT330: .WORD   EMS66,EMS53,EMS175,0
219 073634  075542  077321  076363  EMT331: .WORD   EMS33,EMS103,EMS53,EMS175,0
220 073646  075745  077321  076363  EMT332: .WORD   EMS40,EMS103,EMS53,EMS57,0
221 073660  076323  077112  076523  EMT333: .WORD   EMS51,EMS74,EMS57,EMS115,EMS50,EMS70,0
222 073676  077131  100631  076523  EMT334: .WORD   EMS75,EMS141,EMS57,EMS115,EMS47,EMS73,0
223 073714  077131  101015  101057  EMT335: .WORD   EMS75,EMS150,EMS152,EMS57,EMS115,EMS56,EMS73,0
224 073734  076551  076564  076613  EMT336: .WORD   EMS60,EMS61,EMS62,0
225 073744  077630  077660  075570  EMT337: .WORD   EMS116,EMS117,EMS34,0
226 073754  075570  076363  076375  EMT340: .WORD   EMS34,EMS53,EMS54,EMS111,0
227 073766  076417  075570  000000  EMT341: .WORD   EMS55,EMS34,0
228 073774  076351  077660  076446  EMT342: .WORD   EMS52,EMS117,EMS56,EMS57,0
```

ERROR MESSAGE TABLE

```
    229 074006  076351  077660  076153  EMT343: .WORD   EMS52,EMS117,EMS45,EMS57,0
    230 074020  076351  077660  076100  EMT344: .WORD   EMS52,EMS117,EMS44,EMS57,0
    231 074032  076351  077660  102377  EMT345: .WORD   EMS52,EMS117,EMS221,0
    232 074042  077406  075246  077625  EMT346: .WORD   EMS106,EMS20,EMS115,EMS223,EMS72,0
    233 074056  076351  102054  102447  EMT347: .WORD   EMS52,EMS205,EMS222,EMS206,0
    234 074070  077131  101015  077272  EMT350: .WORD   EMS75,EMS150,EMS102,EMS115,EMS56,EMS73,0
    235 074106  101427  077075  077272  EMT351: .WORD   EMS167,EMS73,EMS102,0
    236 074116  102253  000000          EMT352: .WORD   EMS215,0
    237 074122  102324  102013  102274  EMT353: .WORD   EMS217,EMS203,EMS216,0
    238 074132  102377  075246  000000  EMT354: .WORD   EMS221,EMS20,0
```

ERROR MESSAGE TABLE

```
      1 074140  103004  103610  103665  EHT1:    .WORD   EH1,STSH1,STSH2,STSH4,0
      2 074152  103610  103665  104012  EHT2:    .WORD   STSH1,STSH2,STSH4,0
      3
      4 074162  103023  000000          EHT110:  .WORD   EH110,0
      5 074166  103032  000000          EHT111:  .WORD   EH111,0
      6
      7 074172  103051  000000          EHT114:  .WORD   EH114,0
      8 074176  103100  103610  103665  EHT223:  .WORD   EH223,STSH1,STSH2,STSH4,0
      9 074210  103126  103610  103665  EHT256:  .WORD   EH256,STSH1,STSH2,STSH4,0
     10
     11 074222  103202  103610  103665  EHT336:  .WORD   EH336,STSH1,STSH2,STSH4,0
     12 074234  103241  103610  103665  EHT337:  .WORD   EH337,STSH1,STSH2,STSH4,0
     13 074246  103376  103610  103665  EHT344:  .WORD   EH344,STSH1,STSH2,STSH4,0
     14
     15 074260  103534  000000          EHT353:  .WORD   EH353,0
```

ERROR MESSAGE TABLE

```
      1 074264  104050  104144  104162  EDT1:    .WORD   ED1,STSD1,STSD2,STSD4
      2 074274  104144  104162  104214  EDT2:    .WORD   STSD1,STSD2,STSD4
      3
      4 074302  104056                  EDT110:  .WORD   ED110
      5 074304  104062                  EDT111:  .WORD   ED111
      6
      7 074306  104070                  EDT114:  .WORD   ED114
      8 074310  104100  104144  104162  EDT223:  .WORD   ED223,STSD1,STSD2,STSD4
      9
     10 074320  104110  104144  104162  EDT336:  .WORD   ED336,STSD1,STSD2,STSD4
     11 074330  104122  104144  104162  EDT337:  .WORD   ED337,STSD1,STSD2,STSD4
     12 074340  104122  104144  104162  EDT344:  .WORD   ED337,STSD1,STSD2,STSD4,0
     13
     14 074352  104134                  EDT353:  .WORD   ED353
```

ERROR MESSAGE TABLE

```
      1 074354  104227  104245  104245  EFT1:    .WORD   EF111,STSF,STSF,STSF
      2 074364  104245  104245  104245  EFT2:    .WORD   STSF,STSF,STSF
      3
      4 074372  104226                  EFT110:  .WORD   EF110
      5 074374  104227                  EFT111:  .WORD   EF111
      6
      7 074376  104231                  EFT114:  .WORD   EF114
      8 074400  104231  104245  104245  EFT223:  .WORD   EF114,STSF,STSF,STSF
      9
     10 074410  104234  104245  104245  EFT336:  .WORD   EF336,STSF,STSF,STSF
     11 074420  104234  104245  104245  EFT337:  .WORD   EF336,STSF,STSF,STSF
     12 074430  104234  104245  104245  EFT344:  .WORD   EF336,STSF,STSF,STSF
     13
     14 074440  104231                  EFT353:  .WORD   EF114
```

ERROR MESSAGE STRINGS

```
                                .SBTTL  ERROR MESSAGE STRINGS

074442  127  122  117  EMS1:   .ASCIZ  @WRONG UNIT SELECTED (RMCS2, BITS 0-2) @
074511  104  105  126  EMS2:   .ASCIZ  @DEVICE WENT @
074526  125  116  101  EMS3:   .ASCIZ  @UNAVAILABLE "DVA" (RMCS1, BIT 11) @
074571  116  117  116  EMS4:   .ASCIZ  @NONEXISTENT "NED" (RMCS2, BIT 12) @
074634  103  117  115  EMS5:   .ASCIZ  @COMMAND NOT COMPLETED, @
074664  103  117  116  EMS6:   .ASCIZ  @CONTROLLER NOT READY (RMCS1, BIT 7) @
074731  104  122  111  EMS7:   .ASCIZ  @DRIVE NOT READY "DRY" (RMDS, BIT 7) @
074776  107  117  040  EMS10:  .ASCIZ  @GO NOT RESET "GO" (RMCS1, BIT 0) @
075040  111  116  126  EMS11:  .ASCIZ  @INVALID @
075051  106  125  116  EMS12:  .ASCIZ  @FUNCTION CODE (RMCS1, BITS 1-5) @
075112  115  101  123  EMS13:  .ASCIZ  @MASSBUS @
075123  103  117  116  EMS14:  .ASCIZ  @CONTROL @
075134  102  125  123  EMS15:  .ASCIZ  @BUS PARITY ERROR @
075156  042  115  103  EMS16:  .ASCIZ  @"MCPE" (RMCS1, BIT 13) @
075206  124  122  101  EMS17:  .ASCIZ  @TRANSFER ERROR (RMCS1, BIT 14) @
075246  123  110  117  EMS20:  .ASCIZ  @SHOULD NOT BE SET @
075271  127  117  122  EMS21:  .ASCIZ  @WORD COUNT (RMWC) @
075314  102  125  123  EMS22:  .ASCIZ  @BUS (RMBA) @
075330  042  114  102  EMS23:  .ASCIZ  @"LBT" (RMDS, BIT 10) @
075356  042  101  117  EMS24:  .ASCIZ  @"AOE" (RMER1, BIT 09) @
075405  104  111  123  EMS25:  .ASCIZ  @DISK (RMDA) @
075422  103  131  114  EMS26:  .ASCIZ  @CYLINDER (RMDC) @
075443  101  104  104  EMS27:  .ASCIZ  @ADDRESS @
075454  123  124  101  EMS30:  .ASCIZ  @STATUS @
075464  042  127  114  EMS31:  .ASCIZ  @"WLE" (RMER1, BIT 11) @
075513  042  125  120  EMS32:  .ASCIZ  @"UPE" (RMCS2, BIT 13) @
075542  042  127  103  EMS33:  .ASCIZ  @"WCF" (RMER1, BIT 5) @
075570  127  122  111  EMS34:  .ASCIZ  @WRITE CHECK ERROR-"WCE" (RMCS2, BIT 14) @
075641  042  115  104  EMS35:  .ASCIZ  @"MDPE" (RMCS2, BIT 8) @
075670  042  104  103  EMS36:  .ASCIZ  @"DCK" (RMER1, BIT 15) @
075717  042  105  103  EMS37:  .ASCIZ  @"ECH" (RMER1, BIT 6) @
075745  042  104  114  EMS40:  .ASCIZ  @"DLT" (RMCS2, BIT 15) @
075774  042  115  130  EMS41:  .ASCIZ  @"MXF" (RMCS2, BIT 9) @
076022  042  104  124  EMS42:  .ASCIZ  @"DTE" (RMER1, BIT 12) @
076051  042  110  103  EMS43:  .ASCIZ  @"HCRC" (RMER1, BIT 8) @
076100  110  105  101  EMS44:  .ASCIZ  @HEADER COMPARE ERROR "HCE" (RMER1, BIT 7) @
076153  106  117  122  EMS45:  .ASCIZ  @FORMAT ERROR "FER" (RMER1, BIT 4) @
076216  042  111  101  EMS46:  .ASCIZ  @"IAE" (RMER1, BIT 10) @
076245  042  117  120  EMS47:  .ASCIZ  @"OPI" (RMER1, BIT 13) @
076274  042  123  113  EMS50:  .ASCIZ  @"SKI" (RMER2, BIT 14) @
076323  042  120  111  EMS51:  .ASCIZ  @"PIP" (RMDS, BIT 13) @
076351  124  110  105  EMS52:  .ASCIZ  @THE RM80 @
076363  104  105  124  EMS53:  .ASCIZ  @DETECTED @
076375  101  124  040  EMS54:  .ASCIZ  @AT AN UNEXPECTED @
076417  111  116  103  EMS55:  .ASCIZ  @INCORRECT DATA DURING @
076446  111  116  126  EMS56:  .ASCIZ  @INVALID COMMAND ERROR "IVC" (RMER2, BIT 12) @
076523  104  125  122  EMS57:  .ASCIZ  @DURING DATA TRANSFER @
076551  104  101  124  EMS60:  .ASCIZ  @DATA READ @
076564  104  117  105  EMS61:  .ASCIZ  @DOES NOT COMPARE WITH @
076613  104  101  124  EMS62:  .ASCIZ  @DATA WRITTEN @
076631  042  120  101  EMS63:  .ASCIZ  @"PAR" (RMER1, BIT 3) @
076657  111  123  040  EMS64:  .ASCIZ  @IS INCORRECT @
076675  103  117  115  EMS65:  .ASCIZ  @COMPOSITE ERROR "ERR" (RMDS, BIT 14) @
076743  104  101  124  EMS66:  .ASCIZ  @DATA PARITY ERROR "DPE" (RMER2, BIT 3) @
077013  104  125  122  EMS67:  .ASCIZ  @DURING SEEK COMMAND @
```

ERROR MESSAGE STRINGS

```
 58 077040      111     123     040  EMS70:  .ASCIZ  @IS RESET @
 59 077052      105     122     122  EMS71:  .ASCIZ  @ERRONEOUS @
 60 077065      111     123     040  EMS72:  .ASCIZ  @IS SET @
 61 077075      104     111     104  EMS73:  .ASCIZ  @DID NOT SET @
 62 077112      104     111     104  EMS74:  .ASCIZ  @DID NOT RESET @
 63 077131      114     117     123  EMS75:  .ASCIZ  @LOST @
 64 077137      104     125     122  EMS76:  .ASCIZ  @DURING PACK ACK COMMAND @
 65 077170      042     122     115  EMS77:  .ASCIZ  @"RMR" (RMER1, BIT 2) @
 66 077216      042     111     114  EMS100: .ASCIZ  @"ILR" (RMER1, BIT 1) @
 67 077244      042     111     114  EMS101: .ASCIZ  @"ILF" (RMER1, BIT 0) @
 68 077272      104     125     122  EMS102: .ASCIZ  @DURING SEARCH COMMAND @
 69 077321      105     122     122  EMS103: .ASCIZ  @ERROR @
 70 077330      042     114     102  EMS104: .ASCIZ  @"LBC" (RMER2, BIT 10) @
 71 077357      042     114     123  EMS105: .ASCIZ  @"LSC" (RMER2, BIT 11) @
 72 077406      110     105     101  EMS106: .ASCIZ  @HEADER ERRORS @
 73 077425      102     125     123  EMS110: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
 74 077454      101     104     104  EMS111: .ASCIZ  @ADDRESS @
 75 077465      127     110     105  EMS112: .ASCII  @WHEN READING/WRITING RH REGISTERS @
 76 077527      101     124     040          .ASCIZ  @AT THE FOLLOWING @
 77 077551      124     110     105  EMS113: .ASCIZ  @THE SELECTED DEVICE IS @
 78 077601      116     117     116  EMS114: .ASCIZ  @NONEXISTENT DEVICE @
 79 077625      015     012     000  EMS115: .ASCIZ  <CR><LF>
 80 077630      124     110     105  EMS116: .ASCIZ  @THE MASSBUS CONTROLLER @
 81 077660      106     101     111  EMS117: .ASCIZ  @FAILED TO DETECT @
 82 077702                           EMS120:         ;NOT USED
 83 077702      103     117     116  EMS121: .ASCIZ  @CONTROL STATUS REGISTER 1, RMCS1, @
 84 077745      102     125     123  EMS122: .ASCIZ  @BUS ADDRESS REGISTER, RMBA, @
 85 100002      103     117     116  EMS123: .ASCIZ  @CONTROL STATUS REGISTER 2, RMCS2, @
 86 100045      105     122     122  EMS124: .ASCIZ  @ERROR REGISTER 1, RMER1, @
 87 100077      101     124     124  EMS125: .ASCIZ  @ATTENTION SUMMARY REGISTER, RMAS, @
 88 100142      115     101     111  EMS126: .ASCIZ  @MAINTENANCE REGISTER #1, RMMR #1, @
 89 100205      105     103     103  EMS127: .ASCIZ  @ECC POSITION REGISTER, RMEC1, @
 90 100244      105     103     103  EMS130: .ASCIZ  @ECC PATTERN REGISTER, RMEC2, @
 91 100302      115     101     111  EMS131: .ASCIZ  @MAINTENANCE REGISTER 2, RMMR2, @
 92 100342      116     117     124  EMS132: .ASCIZ  @NOT INITIALIZED BY @
 93 100366      125     116     111  EMS133: .ASCIZ  @UNIBUS INITIALIZE @
 94 100411      103     117     116  EMS134: .ASCIZ  @CONTROLLER CLEAR, I.E.  BIT 5 OF @
 95 100453      122     110     040  EMS135: .ASCIZ  @RH CONTROLLER ERROR CLEAR (RMCS1, BIT 14) @
 96 100526      104     122     111  EMS136: .ASCIZ  @DRIVE CLEAR COMMAND @
 97 100553      104     122     111  EMS137: .ASCIZ  @DRIVE STATUS REGISTER, RMDS @
 98 100610      115     105     104  EMS140: .ASCIZ  @MEDIUM OFF LINE @
 99 100631      042     115     117  EMS141: .ASCIZ  @"MOL" (RMDS, BIT 12) @
100 100657      104     122     111  EMS142: .ASCIZ  @DRIVE FAULT @
101 100674      042     104     126  EMS143: .ASCIZ  @"DVC" (RMER2, BIT 7) @
102 100722      125     116     123  EMS144: .ASCIZ  @UNSAFE @
103 100732      042     125     116  EMS145: .ASCIZ  @"UNS" (RMER1, BIT 14) @
104 100761      123     110     117  EMS146: .ASCIZ  @SHOULD BE SET @
105 101000      117     106     106  EMS147: .ASCIZ  @OFFSET MODE @
106 101015      126     117     114  EMS150: .ASCIZ  @VOLUME VALID @
107 101033      042     117     115  EMS151: .ASCIZ  @"DM" (RMDS, BIT 0) @
108 101057      042     126     126  EMS152: .ASCIZ  @"VV" (RMDS, BIT 6) @
109 101103      120     101     103  EMS153: .ASCIZ  @PACK ACK COMMAND @
110 101125      116     117     124  EMS154: .ASCIZ  @NOT SET BY @
111 101141      117     106     106  EMS155: .ASCIZ  @OFFSET COMMAND @
112 101161      116     117     124  EMS156: .ASCIZ  @NOT RESET BY @
113 101177      122     124     103  EMS157: .ASCIZ  @RTC COMMAND @
114 101214      122     111     120  EMS160: .ASCIZ  @RIP COMMAND @
```

ERROR MESSAGE STRINGS

```
    115 101231      117     106     106  EMS161: .ASCIZ  @OFFSET REGISTER (RMOF) @
    116 101261                           EMS162:         ;<UNUSED>
    117 101261      104     125     122  EMS163: .ASCIZ  @DURING RECALIBRATE @
    118 101305      111     123     040  EMS164: .ASCII  @IS INTERMITTENT OR DRIVE DIDNT DROP ON @
    119 101354      103     131     114          .ASCIZ  @CYLINDER @
    120 101366      125     116     105  EMS165: .ASCIZ  @UNEXPECTED @
    121 101402      122     105     103  EMS166: .ASCIZ  @RECALIBRATE COMMAND @
    122 101427      042     101     124  EMS167: .ASCIZ  @"ATA" (RMDS, BIT15) @
    123 101454      127     110     105  EMS170: .ASCIZ  @WHEN READING REGISTER @
    124 101503      105     122     122  EMS171: .ASCIZ  @ERROR REGISTER #2, RMER2, @
    125 101536      116     117     116  EMS172: .ASCIZ  @NONRECOVERABLE @
    126 101556      122     105     103  EMS173: .ASCIZ  @RECOVERABLE @
    127 101573      104     101     124  EMS174: .ASCIZ  @DATA @
    128 101601      104     125     122  EMS175: .ASCIZ  @DURING WRITE COMMAND @
    129 101627      042     117     120  EMS176: .ASCIZ  @"OPE" (RMER2, BIT 13) @
    130 101656      111     116     040  EMS177: .ASCIZ  @IN WRITE PROTECT @
    131 101700      103     101     116  EMS200: .ASCIZ  @CAN NOT SET @
    132 101715      104     111     101  EMS201: .ASCIZ  @DIAGNOSTIC MODE "DMD" (RMMR1, BIT 0) @
    133 101763      104     125     122  EMS202: .ASCIZ  @DURING DIAGNOSTIC MODE @
    134 102013      111     116     103  EMS203: .ASCIZ  @INCORRECT @
    135 102026      042     127     122  EMS204: .ASCIZ  @"WRL" (RMDS, BIT 11) @
    136 102054      105     130     105  EMS205: .ASCIZ  @EXECUTED @
    137 102066      127     111     124  EMS206: .ASCIZ  @WITH COMP ERROR SET @
    138 102113      042     107     117  EMS207: .ASCIZ  @"GO" (RMCS1, BIT 0) @
    139 102140      127     122     111  EMS210: .ASCIZ  @WRITING @
    140 102151      127     101     123  EMS211: .ASCIZ  @WAS RESET BY @
    141 102167      120     122     117  EMS212: .ASCIZ  @PROGRAM INTERRUPT @
    142 102212      127     101     123  EMS213: .ASCIZ  @WAS NOT GENERATED @
    143 102235      123     105     105  EMS214: .ASCIZ  @SEEK COMMAND @
    144 102253      120     122     117  EMS215: .ASCIZ  @PROGRAM TIMEOUT @
    145 102274      104     125     122  EMS216: .ASCIZ  @DURING LOOK AHEAD TEST @
    146 102324      1 4     117     117  EMS217: .ASCIZ  @LOOK AHEAD REGISTER,RMLA, @
    147 102357      123     105     101  EMS220: .ASCIZ  @SEARCH COMMAND @
    148 102377      102     101     104  EMS221: .ASCIZ  @BAD SECTOR ERROR "BSE" (RMER2, BIT 15) @
    149 102447      101     040     104  EMS222: .ASCIZ  @A DATA TRANSFER COMMAND @
    150 102500      110     105     101  EMS223: .ASCIZ  @HEADER COMPARE INHIBIT "HCI" (RMOF, BIT 10) @
    151 102555      116     117     116  EMS224: .ASCIZ  @NONEXISTENT MEMORY "NEM" (RMCS2, BIT 11) @
    152 102627      123     113     111  EMS225: .ASCIZ  @SKIP SECTOR ERROR @
    153 102652      042     123     123  EMS226: .ASCIZ  @"SSE" (RMER2, BIT 05) @
    154 102701      111     116     110  EMS227: .ASCIZ  @INHIBIT @
    155 102712      042     123     123  EMS230: .ASCIZ  @"SSEI" (RMOF, BIT 09) @
    156 102741      105     116     104  EMS231: .ASCIZ  @END OF TRACK @
    157 102757      101     106     124  EMS232: .ASCIZ  @AFTER DATA TRANSFER @
```

ERROR MESSAGE STRINGS

```
      1 103004      105      130      120  EM1:     .ASCIZ  @EXPCTD  RECEVD@
      2 103023      102      125      123  EM110:   .ASCIZ  @BUSADR@
      3 103032      040      122      115  EM111:   .ASCIZ  @ RMCS2   RMCS1@
      4
      5 103051      122      105      103  EM114:   .ASCIZ  @RECEVD  SNGPRT  DULPRT@
      6 103100      105      130      120  EM223:   .ASCIZ  @EXPCTD  RECEVD   DATA@
      7 103126      105      130      120  EM256:   .ASCII  @EXPCTD  RECEVD  RGSTR@<CRLF>
      8 103154      123      124      101           .ASCIZ  @STATUS  STATUS  INDEX@
      9
     10 103202      1??      104      101  EM336:   .ASCIZ  @GDADRS  GDDATA  BDADRS  BDDATA@
     11 103241      122      115      103  EM337:   .ASCII  @RMCS2   STATUS  FAILING   DATA@<CRLF>
     12 103300       37      137      137           .ASCII  @                            @<CRLF>
     13 103337      105      130      120           .ASCIZ  @EXPCTD--RECEVD  --BIT---ADRESS@
     14
     15 103376      122      115      105  EM344:   .ASCII  @RMER1   STATUS  HEADER  FAILING@<CRLF>
     16 103436      137      137      137           .ASCII  @                 WORD     BIT@<CRLF>
     17 103474      105      130      120           .ASCIZ  @EXPCTD--RECEVD  NUMBER  POSITON@
     18 103534      105      130      120  EM353:   .ASCII  @EXPCTD  RECEVD@<CRLF>
     19 103553      074      103      122           .ASCIZ  @<CRLF> RMLA    RMLA    RMOF @
     20
     21 103610      040      122      115  STSH1:   .ASCII  @ RMCS1  RMCS2   RMDS    RMER1   RMER2@
     22 103655      040      040      040           .ASCIZ  @   RMAS@
     23 103665      040      122      115  STSH2:   .ASCII  @ RMWC    RMBA    RMDA    RMOF    RMDC@
     24 103732      040      040      040           .ASCIZ  @    RMEC1   RMEC2@
     25 103754      040      122      115  STSH3:   .ASCIZ  @ RMDA    RMDC    RMOF    RMLA@
     26 104012      040      122      115  STSH4:   .ASCIZ  @ RMMR1   RMMR2   RMDT    RMSN@
     27                                    .EVEN
```

ERROR MESSAGE STRINGS

```
      1 104050  001140  001142  000000  ED1:    .WORD   $GDDAT,$BDDAT,0
      2 104056  001276  000000          ED110:  .WORD   $BASE,0
      3 104062  001174  001176  000000  ED111:  .WORD   $TMP0,$TMP1,0
      4
      5 104070  001364  001176  001200  ED114:  .WORD   RMDTI,$TMP1,$TMP2,0
      6 104100  001140  001142  001174  ED223:  .WORD   $GDDAT,$BDDAT,$TMP0,0
      7
      8 104110  001134  001140  001136  ED336:  .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
      9
     10 104122  001140  001142  001174  ED337:  .WORD   $GDDAT,$BDDAT,$TMP0,$TMP1,0
     11 104134  001140  001142  001444  ED353:  .WORD   $GDDAT,$BDDAT,RMOF0,0
     12
     13 104144  001336  001346  001350  STSD1:  .WORD   RMCS1I,RMCS2I,RMDSI,RMER1I,RMER2I,RMASI,0
     14 104162  001340  001342  001344  STSD2:  .WORD   RMWCI,RMBAI,RMDAI,RMOFI,RMDCI,RMEC1I
     15 104176  001404  000000          .WORD   RMEC2I,0
     16 104202  001344  001372  001370  STSD3:  .WORD   RMDAI,RMDCI,RMOFI,RMLAI,0
     17 104214  001362  001376  001364  STSD4:  .WORD   RMMR1I,RMMR2I,RMDTI,RMSNI,0
```

ERROR MESSAGE STRINGS

```
      1 104226      000                       EF110:   .BYTE   0
      2 104227      000       000             EF111:   .BYTE   0,0
      3 104231      000       000       000   EF114:   .BYTE   0,0,0
      4 104234      000       000       000   EF336:   .BYTE   0,0,0,0
      5 104240      000       000       000   EF337:   .BYTE   0,0,0,0,0
      6
      7 104245      000       000       000   STSF:    .BYTE   0,0,0,0,0,0,0
      8                                       .EVEN
```

ERROR MESSAGE STRINGS

```
                                        ;STORAGE FOR GENERAL DATA TRANSFERRS
104254                                  BUFFER:
104254                                  BUFONE: .BLKW   258.
105260                                  BUFTWO: .BLKW   258.

                                        ;STORAGE FOR MANUFACTURES 16 BIT MODE BAD SECTOR FILE
106264  000000  000000                  MFGFIL: .WORD   0,0              ;2 HEADER WORDS
106270                                          .BLKW   256.            ;256. WORDS OF DATA
107270  177777                                  .WORD   -1              ;TERMINATOR IF FILE IS FULL

                                        ;STORAGE FOR USERS 16 BIT MODE BAD SECTOR FILE
107272  000000  000000                  USRFIL: .WORD   0,0              ;2 HEADER WORDS
107276                                          .BLKW   256.            ;256. WORDS OF DATA
110276  177777                                  .WORD   -1              ;TERMINATOR IF FILE IS FULL

                                        ;STORAGE FOR SKIP SECTOR FILE
110300  000000  000000                  SSFIL:  .WORD   0,0              ;2 HEADER WORDS
110304                                          .BLKW   28.             ;2 CYLINDERS OF SKIP SECTOR ENTRIES
110374  177777                                  .WORD   -1              ;TERMINATOR

        104254                          .=BUFFER

104254                                  HELP:
104254    200                           .ASCII  <CRLF>
104255    200                           .ASCII  <CRLF>
104256    114     111     123           .ASCII  @LIST OF TESTS@<CRLF>
104274    055     055     055           .ASCII  @-------------@<CRLF>
104312    124     061     011           .ASCII  @T1     CONTROLLER ACCESS TEST@<CRLF>
104344    124     062     011           .ASCII  @T2     WRITE, READ ZEROS@<CRLF>
104371    124     063     011           .ASCII  @T3     WRITE, WRITE CHECK ZEROS@<CRLF>
104425    124     064     011           .ASCII  @T4     WRITE, WRITE CHECK ZEROS W/ WCE ERROR@<CRLF>
104476    124     065     011           .ASCII  @T5     WRITE, READ ONES@<CRLF>
104522    124     066     011           .ASCII  @T6     WRITE, WRITE CHECK ONES@<CRLF>
104555    124     067     011           .ASCII  @T7     WRITE, WRITE CHECK ONES W/ WCE ERROR@<CRLF>
104625    124     061     060           .ASCII  @T10    WRITE, WRITE CHECK MULTIPLE SECTORS@<CRLF>
104675    124     061     061           .ASCII  @T11    WRITE, READ W/ IMPLIED SEEK@<CRLF>
104735    124     061     062           .ASCII  @T12    WRITE, WRITE CHECK W/ HEAD SWITCHING@<CRLF>
105006    124     061     063           .ASCII  @T13    WRITE, WRITE CHECK W/ MID-TRANSFER SEEK@<CRLF>
105062    124     061     064           .ASCII  @T14    WRITE, READ W/ HCE ERROR@<CRLF>
105117    124     061     065           .ASCII  @T15    WRITE, READ W/ HCI & SSEI@<CRLF>
105155    124     061     066           .ASCII  @T16    WRITE, READ W/ IVC ERROR@<CRLF>
105212    124     061     067           .ASCII  @T17    WRITE, READ W/ ABORT@<CRLF>
105243    124     062     060           .ASCII  @T20    WRITE, READ EARLY PEAK SHIFT@<CRLF>
105304    124     062     061           .ASCII  @T21    WRITE, READ EACH TRACK W/ MIXED PEAK SHIFT@<CRLF>
105363    124     062     062           .ASCII  @T22    READ, WRITE CHECK MULTIPLE SECTORS IN OFFSET MODE@<CRLF>
105451    124     062     063           .ASCII  @T23    FORMAT FE CYLINDERS@<CRLF>
105501    200                           .ASCII  <CRLF>
105502    117     120     105           .ASCII  @OPERATIONAL SWITCH SETTINGS@<CRLF>
105536    055     055     055           .ASCII  @-------------------------@<CRLF>
105572    123     127     111           .ASCII  @SWITCH                   USE@<CRLF>
105607    055     055     055           .ASCII  @------          ----------------@<CRLF>
105644    040     040     061           .ASCII  @  15            HALT ON ERROR@<CRLF>
105670    040     040     061           .ASCII  @  14            LOOP ON TEST@<CRLF>
105713    040     040     061           .ASCII  @  13            INHIBIT ERROR TYPEOUTS@<CRLF>
105750    040     040     061           .ASCII  @  12            @<CRLF>
105757    040     040     061           .ASCII  @  11            INHIBIT ITERATIONS@<CRLF>
106010    040     04C     061           .ASCII  @  10            BELL ON ERROR@<CRLF>
```

ERROR MESSAGE STRINGS

```
     58 106034      040      040      040  .ASCII  @   9          LOOP ON ERROR@<CRLF>
     59 106060      040      040      040  .ASCII  @   8          LOOP ON TEST IN SWR<7:0>@<CRLF>
     60 106117      040      040      040  .ASCII  @   7          TN128@<CRLF>
     61 106133      040      040      040  .ASCII  @   6          TN64@<CRLF>
     62 106146      040      040      040  .ASCII  @   5          TN32@<CRLF>
     63 106161      040      040      040  .ASCII  @   4          TN16@<CRLF>
     64 106174      040      040      040  .ASCII  @   3          TN8@<CRLF>
     65 106206      040      040      040  .ASCII  @   2          TN4@<CRLF>
     66 106220      040      040      040  .ASCII  @   1          TN2@<CRLF>
     67 106232      040      040      040  .ASCIZ  @   0          TN1@<CRLF>
     68
     69           000200                   .END    200
```

SYMBOL TABLE

```
ABASE = 176700        ATNMSK= 000377        BUFONE  104254        EBL    = 020000        EH344   103376
ACDW1 = 000000        ATNTBL  067344        BUFTWO  105260        ECH    = 000100        EH353   103534
ACDW2 = 000000        AUNIT = 000000        CC     = 004000        ECI    = 004000        EMS1    074442
ACKSTS  051026        AUSWR = 000000        CH     = 002000        ECRC   = 001000        EMS10   074776
ACPUOP= 000000        AVECT1= 120254        CHGADR  001330        EDT1     074264        EMS100  077216
ADDW0 = 000000        AVECT2= 000000        CHRCNT  063467        EDT110   074302        EMS101  077244
ADDW1 = 000000        A16    = 000400       CKSWR = 104410        EDT111   074304        EMS102  077272
ADDW10= 000000        A17    = 001000       CLKADR  001522        EDT114   074306        EMS103  077321
ADDW11= 000000        BACK   = 000001       CLKVCT  001524        EDT2     074274        EMS104  077330
ADDW12= 000000        BADSCT  034762        CLR    = 000040        EDT223   074310        EMS105  077357
ADDW13= 000000        BADTMO  005344        CLRSTS  050146        EDT336   074320        EMS106  077406
ADDW14= 000000        BAI    = 000010       CMNSTA  007332        EDT337   074330        EMS11   075040
ADDW15= 000000        BB00   = 000001       CMPBUF  037462        EDT344   074340        EMS110  077425
ADDW2 = 000000        BB01   = 000002       CMPERR  046124        EDT353   074352        EMS111  077454
ADDW3 = 000000        BB02   = 000004       CNSL01  066521        ED1      104050        EMS112  077465
ADDW4 = 000000        BB03   = 000010       CNSL02  066531        ED110    104056        EMS113  077551
ADDW5 = 000000        BB04   = 000020       CNSL03  066573        ED111    104062        EMS114  077601
ADDW6 = 000000        BB05   = 000040       CNSL04  066602        ED114    104070        EMS115  077625
ADDW7 = 000000        BB06   = 000100       CNSL07  066636        ED223    104100        EMS116  077630
ADDW8 = 000000        BB07   = 000200       CNSL08  067000        ED336    104110        EMS117  077660
ADDW9 = 000000        BB08   = 000400       CNSL09  067001        ED337    104122        EMS12   075051
ADEVCT= 000000        BB09   = 001000       CNTCLR  050030        ED353    104134        EMS120  077702
ADEVM = 000000        BIT0   = 000001       COMMA   066105        EECC   = 000020        EMS121  077702
ADR    = 000001       BIT00  = 000001       CONT   = 000100        EFT1     074354        EMS122  077745
AENV   = 000000       BIT01  = 000002       CPSAVE  062640        EFT110   074372        EMS123  100002
AENVM = 000000        BIT02  = 000004       CR     = 000015        EFT111   074374        EMS124  100045
AFATAL= 000000        BIT03  = 000010       CRLF   = 000200        EFT114   074376        EMS125  100077
ALL     066074        BIT04  = 000020       CTLFG   001326        EFT2     074364        EMS126  100142
AMADR1= 000000        BIT05  = 000040       CYLMSK= 001777        EFT223   074400        EMS127  100205
AMADR2= 000000        BIT06  = 000100       DBCK   = 100000        EFT336   074410        EMS13   075112
AMADR3= 000000        BIT07  = 000200       DBEN   = 040000        EFT337   074420        EMS130  100244
AMADR4= 000000        BIT08  = 000400       DBL    = 002000        EFT344   074430        EMS131  100302
AMAMS1= 000000        BIT09  = 001000       DCK    = 100000        EFT353   074440        EMS132  100342
AMAMS2= 000000        BIT1   = 000002       DDISP  = 177570        EF110    104226        EMS133  100366
AMAMS3= 000000        BIT10  = 002000       DEBL   = 020000        EF111    104227        EMS134  100411
AMAMS4= 000000        BIT11  = 004000       DEVSEL  046336        EF114    104231        EMS135  100453
AMSGAD= 000000        BIT12  = 010000       DISPLA  001156        EF336    104234        EMS136  100526
AMSGLG= 000000        BIT13  = 020000       DISPRE  000174        EF337    104240        EMS137  100553
AMSGTY= 000000        BIT14  = 040000       DLT    = 100000        EHT1     074140        EMS14   075123
AMTYP1= 000000        BIT15  = 100000       DMD    = 000001        EHT110   074162        EMS140  100610
AMTYP2= 000000        BIT2   = 000004       DPE    = 000010        EHT111   074166        EMS141  100631
AMTYP3= 000000        BIT3   = 000010       DPEHI  = 040000        EHT114   074172        EMS142  100657
AMTYP4= 000000        BIT4   = 000020       DPELO  = 020000        EHT2     074152        EMS143  100674
AOE    = 001000       BIT5   = 000040       DPR    = 000400        EHT223   074176        EMS144  100722
APASS  = 000000       BIT6   = 000100       DRIVES  067176        EHT256   074210        EMS145  100732
APE    = 100000       BIT7   = 000200       DRQ    = 004000        EHT336   074222        EMS146  100761
APRIOR= 000000        BIT8   = 000400       DRVCLR= 000010        EHT337   074234        EMS147  101000
APTCSU= 000040        BIT9   = 001000       DRVSTS  053364        EHT344   074246        EMS15   075134
APTENV= 000001        BLNKS1  067241        DRY    = 000200        EHT353   074260        EMS150  101015
APTSIZ= 000200        BLNKS2  067240        DSWR   = 177570        EH1      103004        EMS151  101033
APTSPO= 000100        BLNKS3  067237        DTASTS  054166        EH110    103023        EMS152  101057
ARGS   = 000004       BLNKS4  067236        DTE    = 010000        EH111    103032        EMS153  101103
ASNDA   001520        BOTADR  063464        DTO    = 010000        EH114    103051        EMS154  101125
ASNDC   001516        BOTFLG  063466        DULPRT= 024026        EH223    103100        EMS155  101141
ASWREG= 000000        BPTVEC= 000014        DVA    = 004000        EH256    103126        EMS156  101161
ATA    = 100000       BSE    = 100000       DVC    = 000200        EH336    103202        EMS157  101177
ATESTN= 000000        BUFFER  104254        EARLY   067670        EH337    103241        EMS16   075156
```

SYMBOL TABLE

EMS160	101214	EMS33	075542	EMT114	071254	EMT177	072142	EMT260	072712		
EMS161	101231	EMS34	075570	EMT115	071256	EMT2	070006	EMT261	072724		
EMS162	101261	EMS35	075641	EMT116	071266	EMT20	070136	EMT262	072744		
EMS163	101261	EMS36	075670	EMT117	071276	EMT200	072144	EMT263	072754		
EMS164	101305	EMS37	075717	EMT12	070062	EMT201	072156	EMT264	072764		
EMS165	101366	EMS4	074571	EMT120	071306	EMT202	072170	EMT265	073004		
EMS166	101402	EMS40	075745	EMT121	071316	EMT203	072202	EMT266	073024		
EMS167	101427	EMS41	075774	EMT122	071326	EMT204	072214	EMT267	073036		
EMS17	075206	EMS42	076022	EMT123	071336	EMT205	072232	EMT27	070226		
EMS170	101454	EMS43	076051	EMT124	071346	EMT206	072242	EMT270	073054		
EMS171	101503	EMS44	076100	EMT125	071356	EMT207	072252	EMT271	073070		
EMS172	101536	EMS45	076153	EMT126	071366	EMT21	070146	EMT272	073106		
EMS173	101556	EMS46	076216	EMT127	071400	EMT210	072264	EMT273	073124		
EMS174	101573	EMS47	076245	EMT13	070070	EMT211	072272	EMT274	073142		
EMS175	101601	EMS5	074634	EMT130	071412	EMT212	072306	EMT275	073160		
EMS176	101627	EMS50	076274	EMT131	071424	EMT213	072322	EMT276	073172		
EMS177	101656	EMS51	076323	EMT132	071436	EMT214	072332	EMT277	073206		
EMS2	074511	EMS52	076351	EMT133	071450	EMT215	072352	EMT3	070014		
EMS20	075246	EMS53	076363	EMT134	071462	EMT216	072364	EMT30	070236		
EMS200	101700	EMS54	076375	EMT135	071474	EMT217	072374	EMT300	073224		
EMS201	101715	EMS55	076417	EMT136	071506	EMT22	070156	EMT301	073244		
EMS202	101763	EMS56	076446	EMT137	071520	EMT220	072404	EMT302	073266		
EMS203	102013	EMS57	076523	EMT14	070102	EMT221	072414	EMT303	073300		
EMS204	102026	EMS6	074664	EMT140	071530	EMT222	072424	EMT304	073312		
EMS205	102054	EMS60	076551	EMT141	071540	EMT223	072434	EMT305	073324		
EMS206	102066	EMS61	076564	EMT142	071550	EMT224	072444	EMT306	073336		
EMS207	102113	EMS62	076613	EMT143	071560	EMT225	072460	EMT307	073346		
EMS21	075271	EMS63	076631	EMT144	071570	EMT226	072500	EMT31	070246		
EMS210	102140	EMS64	076657	EMT145	071600	EMT227	072510	EMT310	073366		
EMS211	102151	EMS65	076675	EMT146	071610	EMT23	070166	EMT311	073400		
EMS212	102167	EMS66	076743	EMT147	071620	EMT230	072524	EMT312	073412		
EMS213	102212	EMS67	077013	EMT15	070110	EMT231	072526	EMT313	073424		
EMS214	102235	EMS7	074731	EMT150	071630	EMT232	072530	EMT314	073444		
EMS215	102253	EMS70	077040	EMT151	071640	EMT233	072532	EMT315	073456		
EMS216	102274	EMS71	077052	EMT152	071652	EMT234	072534	EMT316	073470		
EMS217	102324	EMS72	077065	EMT153	071664	EMT235	072536	EMT317	073502		
EMS22	075314	EMS73	077075	EMT154	071702	EMT236	072540	EMT32	070256		
EMS220	102357	EMS74	077112	EMT155	071720	EMT237	072542	EMT320	073512		
EMS221	102377	EMS75	077131	EMT156	071732	EMT24	070176	EMT321	073522		
EMS222	102447	EMS76	077137	EMT157	071744	EMT240	072544	EMT322	073532		
EMS223	102500	EMS77	077170	EMT16	070116	EMT241	072546	EMT323	073540		
EMS224	102555	EMTVEC=	000030	EMT160	071756	EMT242	072550	EMT324	073550		
EMS225	102627	EMT1	070002	EMT161	071766	EMT243	072552	EMT325	073562		
EMS226	102652	EMT10	070052	EMT162	072000	EMT244	072554	EMT326	073574		
EMS227	102701	EMT100	071056	EMT163	072012	EMT245	072556	EMT327	073612		
EMS23	075330	EMT101	071074	EMT164	072022	EMT246	072560	EMT33	070266		
EMS230	102712	EMT102	071112	EMT165	072030	EMT247	072570	EMT330	073624		
EMS231	102741	EMT103	071122	EMT166	072036	EMT25	070206	EMT331	073634		
EMS232	102757	EMT104	071134	EMT167	072044	EMT250	072602	EMT332	073646		
EMS24	075356	EMT105	071152	EMT17	070126	EMT251	072616	EMT333	073660		
EMS25	075405	EMT106	071162	EMT170	072052	EMT252	072624	EMT334	073676		
EMS26	075422	EMT107	071172	EMT171	072064	EMT253	072632	EMT335	073714		
EMS27	075443	EMT11	070056	EMT172	072074	EMT254	072650	EMT336	073734		
EMS3	074526	EMT110	071212	EMT173	072104	EMT255	072660	EMT337	073744		
EMS30	075454	EMT111	071224	EMT174	072114	EMT256	072670	EMT34	070276		
EMS31	075464	EMT112	071232	EMT175	072126	EMT257	072700	EMT340	073754		
EMS32	075513	EMT113	071240	EMT176	072134	EMT26	070216	EMT341	073766		

SYMBOL TABLE

```
EMT342  073774    ERTY00  063470    ILRG60= 000060    OFD    = 000200    RMAS   = 000016
EMT343  074006    ERTY01  063475    ILRG62= 000062    OFFSET= 000014    RMASI    001354
EMT344  074020    ERTY02  063505    ILRG64= 000064    OM     = 000001    RMASO    001430
EMT345  074032    ERTY03  063514    ILRG66= 000066    ONES     067414    RMBA   = 000004
EMT346  074042    ERTY04  063522    ILRG70= 000070    OPE    = 020000    RMBAE  = 000050
EMT347  074056    ERTY05  063525    ILRG72= 000072    OPI    = 020000    RMBAEI   001406
EMT35   070306    ESRC   = 004000    ILRG74= 000074    OR     = 000200    RMBAEO   001462
EMT350  074070    FER    = 000020    ILRG76= 000076    PACACK= 000022    RMBAI    001342
EMT351  074106    FIND   = 000001    IOTVEC= 000020    PAKACK= 000022    RMBAO    001416
EMT352  074116    FMT16  = 010000    IPCK0  = 000001    PAR    = 000010    RMCS1  = 000000
EMT353  074122    FNCDTB   067244    IPCK1  = 000002    PAT    = 000020    RMCS1I   001336
EMT354  074132    FNCMSK= 000077    IPCK2  = 000004    PDA    = 000400    RMCS1O   001412
EMT36   070316    F0     = 000002    IPCK3  = 000010    PFECH    063532    RMCS2  = 000010
EMT37   070326    F1     = 000004    IR     = 000100    PFECH1   063542    RMCS2I   001346
EMT4    070022    F2     = 000010    IVC    = 010000    PFECH2   063630    RMCS2O   001422
EMT40   070336    F3     = 000020    LBC    = 002000    PFECH3   063646    RMCS3  = 000052
EMT41   070344    F4     = 000040    LBT    = 002000    PFECH4   063654    RMCS3I   001410
EMT42   070354    GENBUF   037214    LF     = 000012    PGE    = 002000    RMCS3O   001464
EMT43   070366    GET      040110    LODEV    067104    PGM    = 001000    RMDA   = 000006
EMT44   070376    GETBUF   001336    LS     = 000004    PHA    = 000200    RMDAI    001344
EMT45   070406    GETINX   001526    LSC    = 004000    PIP    = 020000    RMDAO    001420
EMT46   070416    GETSSF   037070    LST    = 000002    PIRQ   = 177772    RMDB   = 000022
EMT47   070426    GETSTS   040024    LSTRK    001334    PIRQVE= 000240    RMDBI    001360
EMT5    070030    GO     = 000001    MCLK   = 004000    PLFS   = 002000    RMDBO    001434
EMT50   070434    GTSWR  = 104407    MCPE   = 020000    PRIERR   041126    RMDC   = 000034
EMT51   070444    HCE    = 000200    MDF    = 000100    PR0    = 000000    RMDCI    001372
EMT52   070456    HCI    = 002000    MDPE   = 000400    PR1    = 000040    RMDCO    001446
EMT53   070474    HCRC   = 000400    MEDENB   001512    PR2    = 000100    RMDS   = 000012
EMT54   070512    HELP     104254    MFGFIL   106264    PR3    = 000140    RMDSI    001350
EMT55   070522    HT     = 000011    MI     = 000004    PR4    = 000200    RMDSO    001424
EMT56   070534    IAE    = 002000    MIXED    067354    PR5    = 000240    RMDT   = 000026
EMT57   070552    IBSAVE   062642    MOC    = 000400    PR6    = 000300    RMDTI    001364
EMT6    070036    IDXMSK= 000077    MOH    = 020000    PR7    = 000340    RMDTO    001440
EMT60   070562    IE     = 000100    MOL    = 010000    PS     = 177776    RMEC1  = 000044
EMT61   070600    ILF    = 000001    MRD    = 002000    PSEL   = 002000    RMEC1I   001402
EMT62   070620    ILF02  = 000002    MS     = 000040    PSW    = 177776    RMEC1O   001456
EMT63   070634    ILF24  = 000024    MSC    = 000002    PUT      040360    RMEC2  = 000046
EMT64   070644    ILF26  = 000026    MSDRVS   067022    PUTBUF   001412    RMEC2I   001404
EMT65   070664    ILF30  = 000030    MSE    = 100000    PUTINX   001555    RMEC2O   001460
EMT66   070704    ILF32  = 000032    MSER   = 000200    PWRVEC= 000024    RMER1  = 000014
EMT67   070716    ILF34  = 000034    MSGDRV   067036    QUES     066101    RMER1I   001352
EMT7    070044    ILF36  = 000036    MSHELP   066110    RCLSTS   051622    RMER1O   001426
EMT70   070730    ILF40  = 000040    MUR    = 001000    RD     = 000070    RMER2  = 000042
EMT71   070740    ILF42  = 000042    MWD    = 000010    RDCHR  = 104411    RMER2I   001400
EMT72   070750    ILF44  = 000044    MWP    = 000010    RDLIN  = 104412    RMER2O   001454
EMT73   070766    ILF46  = 000046    MXF    = 001000    RDOCT  = 104413    RMHR   = 000036
EMT74   071004    ILF54  = 000054    N        067232    RDY    = 000200    RMHRI    001374
EMT75   071014    ILF56  = 000056    NDTMSK= 115760    READY    007524    RMHRO    001450
EMT76   071030    ILF64  = 000064    NED    = 010000    RECAL  = 000006    RMLA   = 000020
EMT77   071044    ILF66  = 000066    NEM    = 004000    RESREG= 104415    RMLAI    001356
ENRGDT  070002    ILF74  = 000074    NONE     067225    RESVEC= 000010    RMLAO    001432
EQUALS  066072    ILF76  = 000076    NOP    = 000000    REX    = 010000    RMMR1  = 000024
ERR    = 040000    ILR    = 000002    NOTAVL   067136    RG     = 040000    RMMR1I   001362
ERRNMB  063462    ILRG50= 000050    NOTPRS   067121    RGDTPT   067354    RMMR1O   001436
ERROR  = 104000    ILRG52= 000052    NOTRM    067067    RH     = 000072    RMMR2  = 000040
ERRTYP  062644    ILRG54= 000054    NSA    = 100000    RIP    = 000020    RMMR2I   001376
ERRVEC= 000004    ILRG56= 000056    OCC    = 100000    RLEASE= 000012    RMMR2O   001452
```

SYMBOL TABLE

```
RMOF   = 000032        STSD4   104214        TST12   020712        $BELL   001212        $ILLUP  065522
RMOFI    001370        STSF    104245        TST13   021734        $BIN    060460        $INTAG  001151
RMOFO    001444        STSH1   103610        TST14   022546        $CDW1   001302        $ITEMB  001130
RMR    = 000004        STSH2   103665        TST15   024416        $CDW2   001304        $LF     001220
RMSN   = 000030        STSH3   103754        TST16   025720        $CHARC  061464        $LFLG   066003
RMSNI    001366        STSH4   104012        TST17   026740        $CKSWR  064170        $LPADR  001122
RMSNO    001442        SWR     001154        TST2    010076        $CMTAG  001114        $LPERR  001124
RMWC   = 000002        SWREG   000176        TST20   027656        $CM3   = 000000        $MADR1  001254
RMWCI    001340        SW0    = 000001        TST21   030664        $CM4   = 000005        $MADR2  001260
RMWCO    001414        SW00   = 000001        TST22   031712        $CNTLC  065066        $MADR3  001264
RQA    = 100000        SW01   = 000002        TST23   033074        $CNTLG  065100        $MADR4  001270
RQB    = 040000        SW02   = 000004        TST3    011104        $CNTLU  065073        $MAIL   001222
RTC    = 000016        SW03   = 000010        TST4    012062        $CPUOP  001250        $MAMS1  001252
R6     =%000006        SW04   = 000020        TST5    013256        $CRLF   001217        $MAMS2  001256
R7     =%000007        SW05   = 000040        TST6    014264        $DBLK   060676        $MAMS3  001262
SADMSK= 000377         SW06   = 000100        TST7    015242        $DDW0   001306        $MAMS4  001266
SAVREG= 104414         SW07   = 000200        TYPBN  = 104406       $DDW1   001310        $MBADR  001102
SA1    = 000001        SW08   = 000400        TYPDS  = 104405       $DDW2   001312        $MFLG   066002
SA16   = 000020        SW09   = 001000        TYPE   = 104401       $DDW3   001314        $MNEW   065116
SA2    = 000002        SW1    = 000002        TYPOC  = 104402       $DDW4   001316        $MSGAD  001236
SA4    = 000004        SW10   = 002000        TYPON  = 104404       $DDW5   001320        $MSGLG  001240
SA8    = 000010        SW11   = 004000        TYPOS  = 104403       $DDW6   001322        $MSGTY  001222
SC     = 100000        SW12   = 010000        UNS    = 040000       $DDW7   001324        $MSWR   065105
SCOPE  = 000004        SW13   = 020000        UNTMSK= 000007        $DEVCT  001232        $MTYP1  001253
SCTMSG   066006        SW14   = 040000        UNTOFF   067155       $DEVM   001300        $MTYP2  001257
SCTMSK= 003700         SW15   = 100000        UNTON    067166       $DOAGN  033766        $MTYP3  001263
SC0    = 000100        SW2    = 000004        UPE    = 020000       $DTBL   060666        $MTYP4  001267
SC1    = 000200        SW3    = 000010        USE    = 040000       $ENDAD  033756        $MXCNT  062152
SC2    = 000400        SW4    = 000020        USRFIL   107272       $ENDCT  033614        $NULL   001170
SC3    = 001000        SW5    = 000040        U0     = 000001       $ENULL  033772        $NWTST= 000001
SC4    = 002000        SW6    = 000100        U1     = 000002       $ENV    001242        $OCNT   061130
SEARCH= 000030         SW7    = 000200        U2     = 000004       $ENVM   001243        $OMODE  061132
SECERR   041760        SW8    = 000400        VV     = 000100       $EOP    033560        $OVER   062136
SEEK   = 000004        SW9    = 001000        WC     = 000040       $EOPCT  033606        $PASS   001230
SEKSTS   046550        SYSTAT   067044        WCD    = 000050       $EOSP   033522        $PASTM  001106
SHUT     033776        TADMSK= 177400         WCE    = 040000       $ERFLG  001117        $POWER  065530
SHUT2    065130        TAG    = 020000        WCEHI  = 010000       $ERMAX  001131        $PWRDN  065362
SIZCLK   040610        TAGADR= 001114         WCELO  = 004000       $ERROR  062250        $PWRMG  065516
SKI    = 040000        TAP    = 040000        WCF    = 000040       $ERRPC  001132        $PWRUP  065434
SNGPRT= 020026         TA1    = 000400        WCH    = 000052       $ERRTB  001604        $QUES   001216
SSE    = 000040        TA2    = 001000        WD     = 000060       $ERTTL  001126        $RDCHR  064532
SSEI   = 001000        TA4    = 002000        WH     = 000062       $ESCAP  001210        $RDLIN  064622
SSF    = 020000        TA8    = 004000        WLE    = 004000       $ETABL  001242        $RDOCT  065152
SSFENB   001514        TBITVE= 000014         WRL    = 004000       $ETEND  001326        $RDSZ  = 000010
SSFIL    110300        TIMOUT   040732        XSIZ     006364       $FATAL  001224        $RESRE  060350
STACK  = 001100        TKVEC  = 000060        XXDP     001332       $FFLG   066004        $RM80   067062
STANDA   006654        TPVEC  = 000064        Y        067234       $FILLC  001172        $RTNAD  033770
START    005434        TRAPVE= 000034         ZEROS    067456       $FILLS  001171        $SAVRE  060312
START1   005424        TRE    = 040000        $APTHD   001100       $GDADR  001134        $SAVR6  065526
START2   005440        TRTVEC= 000014         $ATYC    065564       $GDDAT  001140        $SCOPE  061470
STCDRV   057606        TST    = 010000        $ATY1    065540       $GET42  033746        $SETUP= 000137
STIMER   041074        TSTNMB   063460        $ATY3    065546       $GTSWR  064260        $STUP  = 177777
STKLMT= 177774         TSTPRP   034032        $ATY4    065556       $GT42P  033742        $SVLAD  062102
STOP     062222        TSTQUE   001466        $AUTOB   001150       $HD    = 000000        $SVPC  = 000210
STSD1    104144        TST1     007676        $BASE    001276       $HIBTS   001100        $SWR   = 167400
STSD2    104162        TST10    016440        $BDADR   001136       $HIOCT   065252        $SWREG   001244
STSD3    104202        TST11    017466        $BDDAT   001142       $ICNT    001120        $SWRMK= 000000
```

SYMBOL TABLE

```
$SW08T  062154          $TKS    001160          $TPS    001164          $TYPE   061134          $VECT1  001272
$TESTN  001226          $TKSRV  063740          $TRAP   065254          $TYPEC  061346          $VECT2  001274
$TIMES  001206          $TMP0   001174          $TRAP2  065314          $TYPEX  061466          $XOFF = 000023
$TKB    001162          $TMP1   001176          $TRP  = 000016          $TYPOC  060732          $XON  = 000021
$TKCNT  063660          $TMP2   001200          $TRPAD  065326          $TYPON  060746          $XTSTR  061512
$TKINT  063670          $TMP3   001202          $TSTM   001104          $TYPOS  060706          $$GET4= 000000
$TKQEN= 063667          $TMP4   001204          $TSTNM  001116          $UNIT   001234          $$SW08= 000024
$TKQIN  063662          $TN   = 000024          $TTYIN  065056          $UNITM  001110          $OFILL  061131
$TKQOU  063664          $TPB    001166          $TYPBN  060406          $USWR   001246          .$X   = 001100
$TKQSR  063666          $TPFLG  001173          $TYPDS  060462

. ABS.  110376      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  62208 WORDS  ( 243 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  70 PAGES
CZRNFA.BIC,CZRNFA/C=CZRNFA.DOC,CZRNFA,[20,0]SYSMAC/M
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

```
$DDW3    6-0#
$DDW4    6-0#
$DDW5    6-0#
$DDW6    6-0#
$DDW7    6-0#
$DEVCT   6-0#
$DEVM    6-0#    10-67*  10-78   10-106*  10-110*  11-63*  11-70*  11-97*  11-98   12-4    12-38   13-29*  14-28*
$DOAGN  14-20    14-20   14-20#
$DTBI   38-1     38-1#
$ENDAD   5-5     10-28   14-20#   42-1
$ENDCT  10-23    14-20#
$ENULL  14-20#
$ENV     6-0#    10-28   40-1     42-1     48-1     48-1
$ENVM    6-0#    10-23   10-71    40-1     40-1     48-1
$EOP    14-20#   14-29
$EOPCT  10-23*   14-20   14-20#
$EOSP   14-9#
$ERFLG   6-0#    41-1    41-1     41-1     41-1     41-1    41-1*   42-1    42-1    42-1*
$ERMAX   6-0#    10-23*  41-1     41-1     41-1     41-1*
$ERROR  10-23    42-1#
$ERRPC   6-0#    42-1    42-1     42-1     42-1*    42-1*   43-46
$ERRTB   8-0#    43-64
$ERTTL   6-0#    14-20   14-20    14-20*   42-1     42-1    42-1*
$ESCAP   6-0#    10-23*  41-1*    42-1     42-1     42-1    42-1
$ETABL   6-0#
$ETEND   5-8      6-0#
$FATAL   6-0#    48-1*
$FFLG   48-1     48-1#   48-1*    48-1*    48-1*
$FILLC   6-0#    40-1    40-1     40-1
$FILLS   6-0#    40-1    40-1
$GDADR   6-0#    13-475* 13-478   13-947*  13-950   18-83*  59-8
$GDDAT   6-0#    13-462* 13-463*  13-473*  13-474*  13-482  13-932* 13-933* 13-945* 13-946* 13-954  13-A17* 13-A33* 13-A34*
        13-A45*  13-A58* 13-B25*  13-B41*  13-B42*  13-B53* 13-B66* 13-C90* 13-D56* 13-E59* 13-E60* 13-F18* 13-F19* 13-F73*
        13-F74*  13-G28* 13-G29*  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* 25-38*  25-51*  25-62*  25-63   25-75*  25-8C*  25-83   25-94*
        25-102*  25-105  25-131*  25-134*  25-135*  25-138  25-150* 25-151* 25-154  25-161* 25-544* 25-551* 25-552* 25-553*
        25-554*  25-558* 25-560   25-622*  25-637*  25-638  25-648* 25-653* 25-656  25-666* 25-679* 25-684  25-696* 25-697*
        25-698*  25-700  25-710*  25-711*  25-713   26-40*  26-54*  28-36*  28-48*  28-68*  28-73   28-92*  28-106* 28-107*
        28-134*  28-147* 28-171*  28-172*  28-186*  28-199* 28-200* 28-215* 28-216* 30-31*  30-32   30-41*  30-57*  30-59
        30-68*   30-81*  30-82    30-91*   30-103*  30-104  30-113* 30-126* 30-127  31-25*  31-26*  31-39*  31-40*  31-58*
        31-59*   31-72*  31-73*   31-86*   31-87*   31-100* 31-101* 31-114* 31-115* 32-33*  32-34*  32-47*  32-48*  32-63*
        32-64*   32-80*  32-81*   32-101*  32-102*  32-118* 32-119* 32-132* 32-133* 32-157* 32-158* 32-173* 32-175* 32-187*
        32-188*  32-203* 32-204*  32-220*  32-221*  32-238* 32-239* 32-252* 32-253* 32-268* 32-269* 33-18*  33-19   33-30*
        33-31    33-40*  33-71*   33-72    33-83*   33-84   33-92*  34-29*  34-30*  34-45*  34-46*  34-65*  34-66*  34-85*
        34-86*   34-114* 34-115*  34-133*  34-134*  34-147* 34-148* 34-161* 34-162* 34-172* 34-191* 34-195  34-209* 34-210*
        34-242*  34-243* 34-258*  34-259*  34-272*  34-273* 34-290* 34-291* 34-303* 34-304* 34-318* 34-319* 34-327* 34-358*
        34-359*  34-371* 34-372*  34-384*  34-385*  34-397* 34-398* 34-419* 34-420* 34-444* 34-445* 34-459* 34-460* 34-472*
        34-473*  34-485* 34-486*  34-498*  34-499*  34-519* 34-520* 34-536* 34-537* 34-552* 34-553* 35-44*  35-45*  35-60*
        35-76*   35-7?*  35-96*   35-97*   35-110*  35-111* 59-1    59-6    59-8    59-10   59-11
$GET42  12-34    12-43   13-30    14-20#
$GT42P  14-20    14-20#
$GTSWR  44-1?    46-1    46-1
$HD      4-488    4-488    4-488
$HIBTS   5-8#
$HIOCT  45-1#    45-1*
$ICNT    6-0#    41-1    41-1     41-1     41-1*    41-1*
```

CROSS REFERENCE TABLE (CREF V04.00)

```
$ILLUP  47-1     47-1     47-1#
$INTAG   6-0#    44-1     44-1     44-1     44-1     44-1*
$ITEMB   6-0#    42-1     42-1     42-1     42-1     42-1     42-1*    42-1*    43-41
$LF      6-0#    40-1     40-1     42-1     42-1     44-1     44-1     44-1
$LFLG   48-1#    48-1*
$LPADR   6-0#    10-23*   41-1     41-1     41-1     41-1*    41-1*    41-1*
$LPERR   6-0#    10-23*   13-84*   13-130*  13-231*  13-277*  13-372*  13-418*  13-549*  13-596*  13-697*  13-744*  13-842*  13-889*
        13-:22*  13-:75*  13-;77*  13-<39*  13->03*  13-?41*  13-@57*  13-A68*  13-C50*  13-D18*  13-E68*  13-F90*  13-G98*  13-H44*
        13-I46*  13-I92*  13-J93*  13-K50*  41-1     41-1     41-1     41-1*    42-1
$MADR1   6-0#
$MADR2   6-0#
$MADR3   6-0#
$MADR4   6-0#
$MAIL    5-8      5-8      6-0#     10-23    10-28    13-2     13-33    13-180   13-322   13-496   13-644   13-789   13-968   13-;23
        13-=09   13->73   13-?96   13-B87   13-D97   13-F29   13-G47   13-H92   13-J43   13-L15   40-1     41-1     42-1
$MAMS1   6-0#
$MAMS2   6-0#
$MAMS3   6-0#
$MAMS4   6-0#
$MBADR   5-8#
$MFLG   48-1     48-1#    48-1*    48-1*
$MNEW   44-1     44-1#
$MSGAD   6-0#    48-1     48-1*
$MSGLG   6-0#    48-1*
$MSGTY   6-0#    48-1     48-1     48-1*    48-1*
$MSWR   44-1     44-1#
$MTYP1   6-0#
$MTYP2   6-0#
$MTYP3   6-0#
$MTYP4   6-0#
$MXCNT  41-1     41-1     41-1     41-1#
$NULL    6-0#    40-1     40-1     40-1
$NWTST  13-2     13-2#    13-2#    13-33    13-33#   13-33#   13-180   13-180#  13-180#  13-322   13-322#  13-322#  13-496   13-496#
        13-496#  13-644   13-644#  13-644#  13-789   13-789#  13-789#  13-968   13-968#  13-968#  13-;23   13-;23#  13-;23#  13-=09
        13-=09#  13-=09#  13->73   13->73#  13->73#  13-?96   13-?96#  13-?96#  13-B87   13-B87#  13-B87#  13-D97   13-D97#  13-D97#
        13-F29   13-F29#  13-F29#  13-G47   13-G47#  13-G47#  13-H92   13-H92#  13-H92#  13-J43   13-J43#  13-J43#  13-L15   13-L15#
        13-L15#
$OCNT   39-1#    39-1*    39-1*
$OMODE  39-1     39-1#    39-1*    39-1*    39-1*    39-1*
$OVER   41-1     41-1     41-1     41-1     41-1#
$PASS    6-0#    10-23*   14-20    14-20    14-20    14-20*   14-20*   41-1     41-1     41-1
$PASTM   5-8#
$POWER  47-1     47-1#
$PWRDN  10-23    47-1     47-1#
$PWRMG  47-1#
$PWRUP  47-1     47-1#
$QUES    6-0#    40-1     40-1     42-1     42-1     44-1     44-1     44-1     44-1
$R2A    46-1
$RDCHR  44-1#    46-1     46-1
$RDDEC  46-1
$RDLIN  44-1#    46-1     46-1
$RDOCT  45-1#    46-1     46-1
$RDSZ   44-1     44-1#
$RESRE  36-1#    46-1
$RM80   10-91    43-28    49-26#
$RTNAD  14-20#
```

CROSS REFERENCE TABLE (CREF V04.00)

$SAVR6	47-1	47-1#	47-1*	47-1*	47-1*									
$SAVRE	36-1#	46-1	46-1											
$SCOPE	10-23	41-1#												
$SETUP	4-785	4-785	4-785	4-785	4-785	4-785	4-785#	4-785#	4-785#	4-785#	4-785#	4-785#	4-785#	10-23
	10-23	10-23	10-23	10-23	10-23	10-23	10-23	10-23	10-23	10-23	10-23	10-28	10-28	10-28
	14-20	14-20	41-1	42-1	42-1	42-1	42-1	44-1	44-1	44-1	44-1	44-1		
$STUP	4-785	4-785	4-785	4-785	4-785	4-785	4-785#	4-785#	4-785#	4-785#	4-785#	4-785#	4-785#	4-785#
	4-785#	4-785#	4-785#	4-785#										
$SVLAD	41-1	41-1#												
$SVPC	5-5	5-5#												
$SW08T	41-1	41-1#												
$SWR	4-477#	4-488	4-489	4-489	4-489	4-489	4-489	4-489	4-489	4-489	6-0	6-0	6-0	10-23
	10-23	10-23	10-23	10-23	13-2	13-33	13-180	13-322	13-496	13-644	13-789	13-968	13-;23	13-=09
	13->73	13-?96	13-B87	13-D97	13-F29	13-G47	13-H92	13-J43	13-L15	14-20	14-20	14-20	14-20	14-20
	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1	41-1
	41-1	41-1	41-1	41-1	41-1	41-1	42-1	42-1	42-1	42-1	42-1	42-1	42-1	42-1
	42-1	42-1	42-1	47-1										
$SWREG	6-0#	10-23												
$SWRMK	4-489	4-489	4-489	4-489	4-489	4-489	4-489	4-489	4-489	41-1	41-1	41-1	41-1	41-1
	41-1	41-1	41-1	41-1	41-1									
$TESTN	6-0#	13-2*	13-33*	13-180*	13-322*	13-496*	13-644*	13-789*	13-968*	13-;23*	13-=09*	13->73*	13-?96*	13-B87*
	13-D97*	13-F29*	13-G47*	13-H92*	13-J43*	13-L15*	41-1*	43-37						
$TIMES	6-0#	10-23*	12-46*	13-L15*	14-20*	41-1	41-1	41-1	41-1*	41-1*				
$TKB	6-0#	40-1	40-1	44-1	44-1	44-1	44-1	44-1	44-1	44-1				
$TKCNT	44-1	44-1	44-1#	44-1*	44-1*	44-1*								
$TKINT	11-4	12-49	44-1	44-1	44-1#									
$TKQEN	44-1	44-1	44-1#											
$TKQIN	44-1	44-1	44-1#	44-1*	44-1*	44-1*	44-1*							
$TKQOU	44-1	44-1	44-1#	44-1*	44-1*	44-1*								
$TKQSR	44-1	44-1	44-1	44-1#										
$TKS	6-0#	40-1	40-1	44-1	44-1	44-1	44-1	44-1	44-1*	44-1*	44-1*	44-1*	44-1*	44-1*
$TKSRV	44-1	44-1#												
$TMP0	6-0#	13-54*	13-201*	13-343*	13-465*	13-517*	13-665*	13-810*	13-935*	13-990*	13-:28*	13-;45*	13-@19*	13-A18*
	13-A20*	13-B26*	13-B28*	13-C10*	13-E13*	13-G68*	13-I14*	13-J63*	17-59	18-85*	20-41*	21-35*	27-29*	27-31
	59-3	59-6	59-10											
$TMP1	6-0#	11-13*	11-14	11-16	11-36*	11-38	11-42	11-50*	11-52	11-56	11-66*	11-67	11-73	11-75
	11-76	11-78	11-83*	11-84	11-87	11-88	11-90	11-95	13-55*	13-202*	13-344*	13-466*	13-518*	13-666*
	13-811*	13-936*	13-991*	13-:29*	13-;46*	13-@20*	13-A21*	13-B29*	13-C11*	13-E14*	13-G69*	13-I15*	13-J64*	17-60
	20-42*	20-43	21-36*	21-37	26-24*	26-25*	26-37	26-39	27-28*	27-37	59-3	59-5	59-10	
$TMP2	6-0#	26-27*	26-28*	26-50	26-53	59-5								
$TMP3	6-0#													
$TMP4	6-0#													
$TN	4-478#	4-488	13-2	13-2	13-2	13-2#	13-33	13-33	13-33	13-33#	13-180	13-180	13-180	13-180#
	13-322	13-322	13-322	13-322#	13-496	13-496	13-496	13-496#	13-644	13-644	13-644	13-644#	13-789	13-789
	13-789	13-789#	13-968	13-968	13-968	13-968#	13-;23	13-;23	13-;23	13-;23#	13-=09	13-=09	13-=09	13-=09#
	13->73	13->73	13->73	13->73#	13-?96	13-?96	13-?96	13-?96#	13-B87	13-B87	13-B87	13-B87#	13-D97	13-D97
	13-D97	13-D97#	13-F29	13-F29	13-F29	13-F29#	13-G47	13-G47	13-G47	13-G47#	13-H92	13-H92	13-H92	13-H92#
	13-J43	13-J43	13-J43	13-J43#	13-L15	13-L15	13-L15	13-L15#	41-1	41-1				
$TPB	6-0#	40-1	40-1	40-1*										
$TPFLG	6-0#	40-1	40-1	40-1										
$TPS	6-0#	40-1	40-1	40-1										
$TRAP	10-23	46-1#												
$TRAP2	46-1	46-1#												
$TRP	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1
	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1
	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1	46-1

CROSS REFERENCE TABLE (CREF V04.00)

```
        46-1    46-1    46-1    46-1    46-1    46-1    46-1    46-1    46-1    46-1    46-1#   46-1#   46-1#   46-1#
        46-1#   46-1#   46-1#   46-1#   46-1#   46-1#   46-1#   46-1#   46-1#   46-1#
$TRPAD  46-1    46-1#
$TSTM    5-8#
$TSTNM   6-0#   12-45*  14-20*  41-1    41-1    41-1    41-1    41-1    41-1*   41-1*   42-1    42-1    42-1    44-4
$TTYIN  44-1    44-1    44-1    44-1    44-1    44-1#
$TYPBN  37-1#   46-1    46-1
$TYPDS  38-1#   46-1    46-1
$TYPE   40-1#   46-1    46-1    48-1
$TYPEC  40-1    40-1    40-1    40-1#   44-1
$TYPEX  40-1    40-1    40-1#
$TYPOC  39-1#   46-1    46-1
$TYPON  39-1    39-1#   46-1
$TYPOS  39-1#   46-1
$UNIT    6-0#   12-51*  12-56   12-62   24-49   43-22
$UNITM   5-8#
$USWR    6-0#
$VECT1   6-0#   11-46   11-56*
$VECT2   6-0#
$XOFF   40-1    40-1
$XON    40-1    40-1    44-1
$XTSTR  41-1#
.$ASTA  48-1    48-1
.$X      5-8     5-8#
A16      4-731#
A17      4-730#
ABASE    4-782#  6-0     6-0
ACDW1    6-0     6-0
ACDW2    6-0     6-0
ACKSTS  15-151   16-136  31-15#
ACPUOP   6-0     6-0
ADDW0    6-0     6-0
ADDW1    6-0     6-0
ADDW10   6-0
ADDW11   6-0
ADDW12   6-0
ADDW13   6-0
ADDW14   6-0
ADDW15   6-0
ADDW2    6-0     6-0
ADDW3    6-0     6-0
ADDW4    6-0     6-0
ADDW5    6-0     6-0
ADDW6    6-0     6-0
ADDW7    6-0     6-0
ADDW8    6-0
ADDW9    6-0
ADEVCT   6-0     6-0
ADEVM    6-0     6-0
ADR     13-2    13-2#   13-2#   13-33   13-33#  13-33#  13-180  13-180#  13-180#  13-322  13-322#  13-322#  13-496  13-496#
        13-496# 13-644  13-644# 13-644# 13-789  13-789# 13-789# 13-968   13-968#  13-968# 13-;23  13-;23#  13-;23# 13-=09
        13-=09# 13-=09# 13->73  13->73# 13->73# 13-?96  13-?96# 13-?96#  13-B87   13-B87# 13-B87# 13-D97  13-D97# 13-D97#
        13-F29  13-F29# 13-F29# 13-G47  13-G47# 13-G47# 13-H92  13-H92#  13-H92#  13-J43  13-J43# 13-J43# 13-L15  13-L15#
        13-L15#
AENV     6-0     6-0
AENVM    6-0     6-0
```

CROSS REFERENCE TABLE (CREF V04.00)

```
AFATAL    6-0       6-0
ALL      11-69     49-5#
AMADR1    6-0       6-0
AMADR2    6-0       6-0
AMADR3    6-0       6-0
AMADR4    6-0       6-0
AMAMS1    6-0       6-0
AMAMS2    6-0       6-0
AMAMS3    6-0       6-0
AMAMS4    6-0       6-0
AMSGAD    6-0       6-0
AMSGLG    6-0       6-0
AMSGTY    6-0       6-0
AMTYP1    6-0       6-0
AMTYP2    6-0       6-0
AMTYP3    6-0       6-0
AMTYP4    6-0       6-0
AOE       4-579#    4-590     25-307    25-322    25-337    25-352    25-356    25-679    25-683    50-77     50-78     50-81     50-82     50-85
          50-86
APASS     6-0       6-0
APE       4-760#
APRIOR    6-0
APTCSU   40-1      48-1#
APTENV   40-1      42-1      48-1      48-1#
APTSIZ   10-23     48-1#
APTSPO   40-1      48-1      48-1#
ARGS     13-41     13-41#    13-52#    13-68#    13-71#    13-73#    13-76#    13-80#    13-93     13-93#    13-110#   13-113#   13-115#   13-118#
         13-122#   13-124#   13-126#   13-139    13-139#   13-155#   13-158#   13-160#   13-163#   13-167#   13-169#   13-171#   13-175#   13-188
         13-188#   13-199#   13-215#   13-218#   13-220#   13-223#   13-227#   13-240    13-240#   13-257#   13-260#   13-262#   13-265#   13-269#
         13-271#   13-273#   13-286    13-286#   13-301#   13-304#   13-306#   13-309#   13-313#   13-315#   13-317#   13-330    13-330#   13-341#
         13-356#   13-359#   13-361#   13-364#   13-368#   13-381    13-381#   13-398#   13-401#   13-403#   13-406#   13-410#   13-412#   13-414#
         13-428    13-428#   13-443#   13-446#   13-448#   13-451#   13-455#   13-460#   13-486#   13-504    13-504#   13-515#   13-533#   13-536#
         13-538#   13-541#   13-545#   13-559    13-559#   13-576#   13-579#   13-581#   13-584#   13-588#   13-590#   13-592#   13-605    13-605#
         13-621#   13-624#   13-626#   13-629#   13-633#   13-635#   13-637#   13-639#   13-652    13-652#   13-663#   13-681#   13-684#   13-686#
         13-689#   13-693#   13-707    13-707#   13-724#   13-727#   13-729#   13-732#   13-736#   13-738#   13-740#   13-753    13-753#   13-768#
         13-771#   13-773#   13-776#   13-780#   13-782#   13-784#   13-797    13-797#   13-808#   13-826#   13-829#   13-851#   13-834#   13-838#
         13-852    13-852#   13-869#   13-872#   13-874#   13-877#   13-881#   13-883#   13-885#   13-899    13-899#   13-914#   13-917#   13-919#
         13-922#   13-926#   13-930#   13-960#   13-985    13-985#   13-988#   13-:06#   13-:09#   13-:11#   13-:14#   13-:18#   13-:40    13-:40#
         13-:55#   13-:58#   13-:60#   13-:63#   13-:67#   13-:69#   13-:71#   13-:84    13-:84#   13-:99#   13-;02#   13-;04#   13-;07#   13-;11#
         13-;13#   13-;15#   13-;31    13-;31#   13-;43#   13-;61#   13-;64#   13-;66#   13-;69#   13-;73#   13-;86    13-;86#   13-;91#   13-;94#
         13-;96#   13-;99#   13-<03#   13-<21#   13-<24#   13-<27#   13-<31#   13-<33#   13-<35#   13-<48    13-<48#   13-<53#   13-<56#   13-<58#
         13-<61#   13-<65#   13-<83#   13-<86#   13-<89#   13-<93#   13-<95#   13-<97#   13-<99#   13-=33    13-=33#   13-=48#   13-=51#   13-=53#
         13-=56#   13-=60#   13-=78#   13-=81#   13-=83#   13-=86#   13-=95#   13-=97#   13-=99#   13->12    13->12#   13->16#   13->19#   13->21#
         13->24#   13->28#   13->43#   13->46#   13->48#   13->51#   13->60#   13->62#   13->64#   13->98    13->98#   13-?16#   13-?19#   13-?21#
         13-?24#   13-?33#   13-?35#   13-?37#   13-?50    13-?50#   13-?65#   13-?68#   13-?70#   13-?73#   13-?82#   13-?84#   13-?86#   13-@06
         13-@06#   13-@17#   13-@41#   13-@44#   13-@46#   13-@49#   13-@53#   13-@66    13-@66#   13-@83#   13-@86#   13-@88#   13-@91#   13-@95#
         13-A14#   13-A30#   13-A43#   13-A55#   13-A64#   13-A76    13-A76#   13-A93#   13-A96#   13-A98#   13-B01#   13-B05#   13-B22#   13-B38#
         13-B51#   13-B63#   13-B72#   13-B97    13-B97#   13-C08#   13-C34#   13-C37#   13-C39#   13-C42#   13-C46#   13-C59    13-C59#   13-C77#
         13-C80#   13-C82#   13-C85#   13-C89#   13-D12#   13-D14#   13-D27    13-D27#   13-D43#   13-D46#   13-D48#   13-D51#   13-D55#   13-D78#
         13-D80#   13-D82#   13-E11#   13-E18    13-E18#   13-E39#   13-E42#   13-E44#   13-E47#   13-E51#   13-E56#   13-E64#   13-E77    13-E77#
         13-E99#   13-F02#   13-F04#   13-F07#   13-F11#   13-F16#   13-F24#   13-F44    13-F44#   13-F63#   13-F66#   13-F69#   13-F79#   13-F82#
         13-F84#   13-F86#   13-F99    13-F99#   13-G18#   13-G21#   13-G24#   13-G35#   13-G38#   13-G40#   13-G42#   13-G55    13-G55#   13-G66#
         13-G82#   13-G85#   13-G87#   13-G90#   13-G94#   13-H07    13-H07#   13-H24#   13-H27#   13-H29#   13-H32#   13-H36#   13-H38#   13-H40#
         13-H53    13-H53#   13-H69#   13-H72#   13-H74#   13-H77#   13-H81#   13-H83#   13-H85#   13-H87#   13-I00    13-I00#   13-I12#   13-I30#
         13-I33#   13-I35#   13-I38#   13-I42#   13-I55    13-I55#   13-I72#   13-I75#   13-I77#   13-I80#   13-I84#   13-I86#   13-I88#   13-J01
         13-J01#   13-J17#   13-J20#   13-J22#   13-J25#   13-J29#   13-J31#   13-J33#   13-J35#   13-J50    13-J50#   13-J61#   13-J77#   13-J80#
```

CROSS REFERENCE TABLE (CREF V04.00)

	13-J82#	13-J85#	13-J89#	13-K02	13-K02#	13-K14#	13-K20#	13-K23#	13-K30#	13-K33#	13-K35#	13-K38#	13-K42#	13-K44#
	13-K46#	13-K58	13-K58#	13-K71#	13-K80#	13-K83#	13-K91#	13-K94#	13-K96#	13-K99#	13-L03#	13-L05#	13-L07#	13-L26#
	13-L40	13-L40#	13-L54#	13-L57#	13-L59#	13-L61#	13-L65#	13-L67#						
ASNDA	7-0#	16-307*	16-344	16-374*	16-377	16-380	16-382*	16-383*	16-384	16-386*	16-425			
ASNDC	7-0#	16-306*	16-343	16-387*	16-388	16-424								
ASWREG	6-0	6-0												
ATA	4-559#	25-134	25-135	25-137	28-161	28-162	28-197	28-200	32-145	32-146	32-185	32-188	50-58	50-59
	50-60	50-63	50-64	50-67	50-68	50-69	50-70	50-71	50-72	50-73	50-74	50-75	50-76	50-79
	50-80	50-83	50-84	50-87	50-88									
ATESTN	6-0	6-0												
ATNMSK	4-596#	33-55												
ATNTBL	10-78	10-106	10-110	11-97	12-17	51-3#								
AUNIT	6-0	6-0												
AUSWR	6-0	6-0												
AVECT1	4-783#	6-0	6-0											
AVECT2	6-0	6-0												
BACK	13-41	13-41#	13-52	13-52#	13-68	13-68#	13-71#	13-73#	13-76	13-76#	13-80	13-80#	13-80#	13-93
	13-93#	13-110	13-110#	13-113#	13-115#	13-118	13-118#	13-122	13-122#	13-122#	13-124	13-124#	13-124#	13-126
	13-126#	13-126#	13-139	13-139#	13-155	13-155#	13-158#	13-160#	13-163	13-163#	13-167	13-167#	13-167#	13-169
	13-169#	13-169#	13-171	13-171#	13-171#	13-175	13-175#	13-188	13-188#	13-199	13-199#	13-215	13-215#	13-218#
	13-220#	13-223	13-223#	13-227	13-227#	13-227#	13-240	13-240#	13-257	13-257#	13-260#	13-262#	13-265	13-265#
	13-269	13-269#	13-269#	13-271	13-271#	13-271#	13-273	13-273#	13-273#	13-286	13-286#	13-301	13-301#	13-304#
	13-306#	13-309	13-309#	13-313	13-313#	13-313#	13-315	13-315#	13-315#	13-317	13-317#	13-317#	13-330	13-330#
	13-341	13-341#	13-356	13-356#	13-359#	13-361#	13-364	13-364#	13-368	13-368#	13-368#	13-381	13-381#	13-398
	13-398#	13-401#	13-403#	13-406	13-406#	13-410	13-410#	13-410#	13-412	13-412#	13-412#	13-414	13-414#	13-414#
	13-428	13-428#	13-443	13-443#	13-446#	13-448#	13-451	13-451#	13-455	13-455#	13-455#	13-460	13-460#	13-460#
	13-488	13-488#	13-488#	13-504	13-504#	13-515	13-515#	13-533	13-533#	13-536#	13-538#	13-541	13-541#	13-545
	13-545#	13-545#	13-559	13-559#	13-576	13-576#	13-579#	13-581#	13-584	13-584#	13-588	13-588#	13-588#	13-590
	13-590#	13-590#	13-592	13-592#	13-592#	13-605	13-605#	13-621	13-621#	13-624#	13-626#	13-629	13-629#	13-633
	13-633#	13-633#	13-635	13-635#	13-635#	13-637	13-637#	13-637#	13-639	13-639#	13-652	13-652#	13-663	13-663#
	13-681	13-681#	13-684#	13-686#	13-689	13-689#	13-693	13-693#	13-693#	13-707	13-707#	13-724	13-724#	13-727#
	13-729#	13-732	13-732#	13-736	13-736#	13-736#	13-738	13-738#	13-738#	13-740	13-740#	13-740#	13-753	13-753#
	13-768	13-768#	13-771#	13-773#	13-776	13-776#	13-780	13-780#	13-780#	13-782	13-782#	13-782#	13-784	13-784#
	13-784#	13-797	13-797#	13-808	13-808#	13-826	13-826#	13-829#	13-831#	13-834	13-834#	13-838	13-838#	13-838#
	13-852	13-852#	13-869	13-869#	13-872#	13-874#	13-877	13-877#	13-881	13-881#	13-881#	13-883	13-883#	13-883#
	13-885	13-885#	13-885#	13-899	13-899#	13-914	13-914#	13-917#	13-919#	13-922	13-922#	13-926	13-926#	13-926#
	13-930	13-930#	13-930#	13-960	13-960#	13-960#	13-985	13-985#	13-988	13-988#	13-:06	13-:06#	13-:09#	13-:11#
	13-:14	13-:14#	13-:18	13-:18#	13-:18#	13-:40	13-:40#	13-:55	13-:55#	13-:58#	13-:60#	13-:63	13-:63#	13-:67
	13-:67#	13-:67#	13-:69	13-:69#	13-:69#	13-:71	13-:71#	13-:71#	13-:84	13-:84#	13-:99	13-:99#	13-;02#	13-;04#
	13-;07	13-;07#	13-;11	13-;11#	13-;11#	13-;13	13-;13#	13-;13#	13-;15	13-;15#	13-;15#	13-;31	13-;31#	13-;43
	13-;43#	13-;61	13-;61#	13-;64#	13-;66#	13-;69	13-;69#	13-;73	13-;73#	13-;73#	13-;86	13-;86#	13-;91	13-;91#
	13-;94#	13-;96#	13-;99	13-;99#	13-<03	13-<03#	13-<03#	13-<21	13-<21#	13-<24#	13-<27	13-<27#	13-<31	13-<31#
	13-<31#	13-<33	13-<33#	13-<33#	13-<35	13-<35#	13-<35#	13-<48	13-<48#	13-<53	13-<53#	13-<56#	13-<58#	13-<61
	13-<61#	13-<65	13-<65#	13-<65#	13-<83	13-<83#	13-<86#	13-<89	13-<89#	13-<93	13-<93#	13-<93#	13-<95	13-<95#
	13-<95#	13-<97	13-<97#	13-<97#	13-<99	13-<99#	13-=33	13-=33#	13-=48	13-=48#	13-=51#	13-=53#	13-=56	13-=56#
	13-=60	13-=60#	13-=60#	13-=78	13-=78#	13-=81#	13-=83#	13-=86	13-=86#	13-=95	13-=95#	13-=95#	13-=97	13-=97#
	13-=97#	13-=99	13-=99#	13-=99#	13->12	13->12#	13->16	13->16#	13->19#	13->21#	13->24	13->24#	13->28	13->28#
	13->28#	13->43	13->43#	13->46#	13->48#	13->51	13->51#	13->60	13->60#	13->60#	13->62	13->62#	13->62#	13->64
	13->64#	13->64#	13->98	13->98#	13-?16	13-?16#	13-?19#	13-?21#	13-?24	13-?24#	13-?33	13-?33#	13-?33#	13-?35
	13-?35#	13-?35#	13-?37	13-?37#	13-?37#	13-?50	13-?50#	13-?65	13-?65#	13-?68#	13-?70#	13-?73	13-?73#	13-?82
	13-?82#	13-?82#	13-?84	13-?84#	13-?84#	13-?86	13-?86#	13-?86#	13-a06	13-a06#	13-a17	13-a17#	13-a41	13-a41#
	13-a44#	13-a46#	13-a49	13-a49#	13-a53	13-a53#	13-a53#	13-a66	13-a66#	13-a83	13-a83#	13-a86#	13-a88#	13-a91
	13-a91#	13-a95	13-a95#	13-a95#	13-A14	13-A14#	13-A14#	13-A30	13-A30#	13-A30#	13-A43	13-A43#	13-A43#	13-A55
	13-A55#	13-A55#	13-A64	13-A64#	13-A64#	13-A76	13-A76#	13-A93	13-A93#	13-A96#	13-A98#	13-B01	13-B01#	13-B05
	13-B05#	13-B05#	13-B22	13-B22#	13-B22#	13-B38	13-B38#	13-B38#	13-B51	13-B51#	13-B51#	13-B63	13-B63#	13-B63#
	13-B72	13-B72#	13-B72#	13-B97	13-B97#	13-C08	13-C08#	13-C34	13-C34#	13-C37#	13-C39#	13-C42	13-C42#	13-C46

CROSS REFERENCE TABLE (CREF V04.00)

	13-C46#	13-C46#	13-C59	13-C59#	13-C77	13-C77#	13-C80#	13-C82#	13-C85	13-C85#	13-C89	13-C89#	13-C89#	13-D12
	13-D12#	13-D12#	13-D14	13-D14#	13-D14#	13-D27	13-D27#	13-D43	13-D43#	13-D46#	13-D48#	13-D51	13-D51#	13-D55
	13-D55#	13-D55#	13-D78	13-D78#	13-D78#	13-D80	13-D80#	13-D80#	13-D82	13-D82#	13-E11	13-E11#	13-E18	13-E18#
	13-E39	13-E39#	13-E42#	13-E44#	13-E47	13-E47#	13-E51	13-E51#	13-E51#	13-E56	13-E56#	13-E56#	13-E64	13-E64#
	13-E64#	13-E77	13-E77#	13-E99	13-E99#	13-F02#	13-F04#	13-F07	13-F07#	13-F11	13-F11#	13-F11#	13-F16	13-F16#
	13-F16#	13-F24	13-F24#	13-F24#	13-F44	13-F44#	13-F63	13-F63#	13-F66#	13-F69	13-F69#	13-F79#	13-F82	13-F82#
	13-F84	13-F84#	13-F84#	13-F86	13-F86#	13-F86#	13-F99	13-F99#	13-G18	13-G18#	13-G21#	13-G24	13-G24#	13-G35#
	13-G38	13-G38#	13-G40	13-G40#	13-G40#	13-G42	13-G42#	13-G42#	13-G55	13-G55#	13-G66	13-G66#	13-G82	13-G82#
	13-G85#	13-G87#	13-G90	13-G90#	13-G94	13-G94#	13-G94#	13-H07	13-H07#	13-H24	13-H24#	13-H27#	13-H29#	13-H32
	13-H32#	13-H36	13-H36#	13-H36#	13-H38	13-H38#	13-H38#	13-H40	13-H40#	13-H40#	13-H53	13-H53#	13-H69	13-H69#
	13-H72#	13-H74#	13-H77	13-H77#	13-H81	13-H81#	13-H81#	13-H83	13-H83#	13-H83#	13-H85	13-H85#	13-H85#	13-H87
	13-H87#	13-I00	13-I00#	13-I12	13-I12#	13-I30	13-I30#	13-I33#	13-I35#	13-I38	13-I38#	13-I42	13-I42#	13-I42#
	13-I55	13-I55#	13-I72	13-I72#	13-I75#	13-I77#	13-I80	13-I80#	13-I84	13-I84#	13-I84#	13-I86	13-I86#	13-I86#
	13-I88	13-I88#	13-I88#	13-J01	13-J01#	13-J17	13-J17#	13-J20#	13-J22#	13-J25	13-J25#	13-J29	13-J29#	13-J29#
	13-J31	13-J31#	13-J31#	13-J33	13-J33#	13-J33#	13-J35	13-J35#	13-J50	13-J50#	13-J61	13-J61#	13-J77	13-J77#
	13-J80#	13-J82#	13-J85	13-J85#	13-J89	13-J89#	13-J89#	13-K02	13-K02#	13-K14	13-K14#	13-K20	13-K20#	13-K23#
	13-K30	13-K30#	13-K33#	13-K35#	13-K38	13-K38#	13-K42	13-K42#	13-K42#	13-K44	13-K44#	13-K44#	13-K46	13-K46#
	13-K46#	13-K58	13-K58#	13-K71	13-K71#	13-K80	13-K80#	13-K83#	13-K91	13-K91#	13-K94#	13-K96#	13-K99	13-K99#
	13-L03	13-L03#	13-L03#	13-L05	13-L05#	13-L05#	13-L07	13-L07#	13-L07#	13-L26	13-L26#	13-L40	13-L40#	13-L54
	13-L54#	13-L57#	13-L59#	13-L61	13-L61#	13-L65	13-L65#	13-L65#	13-L67	13-L67#	13-L67#			
BADSCT	13-52	13-199	13-341	13-515	13-663	13-808	13-988	13-;43	13-a17	13-C08	13-E11	13-G66	13-I12	13-J61
	13-L26	16-33#												
BADTM)	10-3#	10-25												
BAI	4-749#	25-556												
B800	4-678#													
B801	4-678#													
B802	4-678#													
B803	4-678#													
B804	4-678#													
B805	4-678#													
B806	4-678#													
B807	4-678#													
B808	4-678#													
B809	4-678#													
BIT0	4-492#	23-43												
BIT00	4-492	4-492#	4-504	4-551	4-569	4-588	4-625	4-643	4-678	4-752	4-770	41-1	41-1	42-1
	42-1													
BIT01	4-492	4-492#	4-503	4-550	4-587	4-624	4-642	4-678	4-752	4-769				
BIT02	4-492	4-492#	4-502	4-549	4-586	4-623	4-641	4-678	4-752	4-768				
BIT03	4-492	4-492#	4-501	4-548	4-585	4-622	4-640	4-678	4-690	4-749	4-767			
BIT04	4-492	4-492#	4-500	4-547	4-584	4-639	4-678	4-748						
BIT05	4-492	4-492#	4-499	4-583	4-621	4-638	4-678	4-689	4-747					
BIT06	4-492	4-492#	4-568	4-582	4-604	4-620	4-637	4-678	4-733	4-746	4-766			
BIT07	4-492	4-492#	4-567	4-581	4-603	4-619	4-636	4-661	4-678	4-688	4-732	4-745		
BIT08	4-492	4-492#	4-544	4-566	4-580	4-602	4-618	4-635	4-678	4-731	4-744	41-1		
BIT09	4-492	4-492#	4-543	4-565	4-579	4-601	4-617	4-634	4-660	4-678	4-730	4-743	41-1	41-1
	42-1	42-1												
BIT1	4-492#	16-312	17-26	25-576										
BIT10	4-492#	4-542	4-564	4-578	4-600	4-616	4-633	4-659	4-675	4-687	4-729	4-742	4-765	15-165
	42-1													
BIT11	4-492#	4-498	4-541	4-563	4-577	4-615	4-632	4-650	4-658	4-674	4-686	4-741	4-764	15-167
	41-1													
BIT12	4-492#	4-562	4-576	4-614	4-631	4-657	4-673	4-685	4-740	4-763	15-111			
BIT13	4-492#	4-561	4-575	4-613	4-630	4-649	4-672	4-684	4-696	4-728	4-739	4-762	42-1	
BIT14	4-492#	4-560	4-574	4-612	4-629	4-648	4-671	4-683	4-695	4-727	4-738	4-761	15-73	41-1
BIT15	4-492#	4-559	4-573	4-611	4-628	4-647	4-670	4-682	4-694	4-726	4-737	4-760	18-33	18-90

CROSS REFERENCE TABLE (CREF V04.00)

```
BIT2      4-492#  23-43
BIT3      4-492#  15-198  34-324  34-411
BIT4      4-492#  15-140  34-430
BIT5      4-492#
BIT6      4-492#  15-87
BIT7      4-492#  10-71   23-20
BIT8      4-492#
BIT9      4-492#
BLNKS1    49-40#
BLNKS2    10-121  11-34   11-48   49-39#
BLNKS3    49-38#
BLNKS4    10-82   49-37#
BOTADR    43-73*  43-91*  43-94   43-109  43-153#
BOTFLG    43-54*  43-101* 43-104  43-107* 43-154#
BPTVEC     4-492#
BSE        4-682# 13-A52  13-A58  13-B60  13-B66  13-C99  13-D01  13-D65  13-D67  16-224  25-427
BUFFER    16-48*  16-292  61-2#   61-21
BUFONE    13-47   13-98   13-175  13-194  13-245  13-336  13-387  13-510  13-564  13-639  13-658  13-712  13-803  13-858
          13-980  13-:26  13-:38  13-<10  13-<99  13-=20  13->86  13-a12  13-a24  13-a26* 13-a28* 13-a71  13-C03  13-C15
          13-C17* 13-C19* 13-C67  13-D82  13-E06  13-F37  13-G61  13-H13  13-H87  13-I07  13-I60  13-J35  13-J56  13-J97
          13-L23  13-L36* 61-3#
BUFTWO    13-144  13-175  13-425* 13-466  13-473  13-475  13-490* 13-610  13-639  13-896* 13-936  13-945  13-947  13-962*
          13-<73  13-<99  13-A81  13-D33  13-D82  13-E87  13-H59  13-H87  13-J06  13-J35  16-60   16-445  16-472  61-4#
CC         4-674#
CH         4-675#
CHGADR     7-0#   10-14*  10-17*  11-25   11-27*
CHRCNT    43-55*  43-78*  43-84*  43-85   43-88*  43-92   43-97*  43-108* 43-155#
CKSWR     41-1    42-1    42-1    46-1#
CLKADR     7-0#   22-6*   22-8    22-12*  22-14   23-20   23-39
CLKVCT     7-0#   22-7*   22-13*
CLR        4-747# 10-84   12-55   29-16
CLRSTS    15-98   30-22#
CMNSTA    10-130  12-2#
CMPBUF    13-175  13-639  13-<99  13-D82  13-H87  13-J35  18-17#
CMPERR    26-21#
CNSL01    11-32   49-15#
CNSL02    11-40   49-16#
CNSL03    11-44   49-17#
CNSL04    11-54   49-18#
CNSL07    11-61   49-19#
CNSL08    11-92   49-21#
CNSL09    49-22#
CNTCLR    15-76   29-12#
COMMA     11-86   12-6    12-14   49-7#
CONT       4-637#
CPSAVE    41-1    41-1    41-1    41-1*   41-1*   42-1    42-1    42-1    42-1    42-1    42-1#   42-1*   42-1*   43-172
CR         4-492# 11-73   11-84   40-1    40-1    43-76   57-79
CRLF       4-492# 10-6    10-28   10-28   10-42   10-53   10-59   12-30   40-1    40-1    49-3    49-5    49-8    49-9
          49-10   49-11   49-12   49-13   49-14   49-15   49-16   49-18   49-19   49-20   49-21   49-22   49-23   49-25
          49-33   58-7    58-11   58-12   58-15   58-16   58-18   61-24   61-25   61-26   61-27   61-28   61-29   61-30
          61-31   61-32   61-33   61-34   61-35   61-36   61-37   61-38   61-39   61-40   61-41   61-42   61-43   61-44
          61-45   61-46   61-47   61-48   61-49   61-50   61-51   61-52   61-53   61-54   61-55   61-56   61-57   61-58
          61-59   61-60   61-61   61-62   61-63   61-64   61-65   61-66   61-67
CTLFG      7-0#   12-47*  14-22   44-3*
CYLMSK     4-665#
DBCK       4-611#
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

	8-705	8-708	8-711	8-714	8-723	55-1#									
EDT110	8-221	55-4#													
EDT111	8-224	8-227	55-5#												
EDT114	55-7#														
EDT2	8-426	8-429	8-513	3-516	55-2#										
EDT223	8-447	8-528	55-8#												
EDT336	8-351	8-681	8-687	8-690	55-10#										
EDT337	8-684	55-11#													
EDT344	8-699	55-12#													
EDT353	8-720	55-14#													
EECC	4-639#														
EF110	56-4	60-1#													
EF111	56-1	56-5	60-2#												
EF114	56-7	56-8	56-14	60-3#											
EF336	56-10	56-11	56-12	60-4#											
EF337	60-5#														
EFT1	8-3	8-6	8-9	8-12	8-15	8-18	8-24	8-27	8-30	8-33	8-36	8-39	8-42	8-45	
	8-48	8-51	8-54	8-57	8-60	8-63	8-66	8-69	8-72	8-75	8-78	8-81	8-84	8-87	
	8-90	8-93	8-96	8-99	8-102	8-105	8-108	8-111	8-114	8-117	8-120	8-123	8-126	8-130	
	8-133	8-136	8-139	8-142	8-146	8-150	8-154	8-157	8-160	8-163	8-166	8-169	8-172	8-175	
	8-178	8-182	8-185	8-188	8-191	8-194	8-197	8-200	8-203	8-206	8-209	8-212	8-215	8-218	
	8-236	8-239	8-242	8-245	8-248	8-251	8-254	8-257	8-260	8-263	8-266	8-269	8-272	8-275	
	8-278	8-281	8-284	8-287	8-290	8-293	8-296	8-299	8-302	8-305	8-308	8-311	8-314	8-317	
	8-320	8-323	8-326	8-329	8-332	8-335	8-338	8-341	8-344	8-347	8-354	8-357	8-360	8-363	
	8-366	8-369	8-372	8-375	8-378	8-381	8-384	8-390	8-393	8-396	8-399	8-402	8-405	8-408	
	8-411	8-414	8-417	8-420	8-423	8-432	8-435	8-438	8-441	8-444	8-450	8-453	8-456	8-459	
	8-504	8-507	8-510	8-519	8-522	8-525	8-531	8-534	8-537	8-540	8-543	8-546	8-549	8-552	
	8-556	8-559	8-563	8-566	8-569	8-573	8-577	8-581	8-586	8-590	8-594	8-597	8-600	8-603	
	8-606	8-609	8-612	8-615	8-618	8-621	8-624	8-627	8-630	8-633	8-636	8-639	8-642	8-645	
	8-648	8-651	8-654	8-657	8-660	8-663	8-666	8-669	8-672	8-675	8-678	8-693	8-696	8-702	
	8-705	8-708	8-711	8-714	8-723	56-1#									
EFT110	8-221	56-4#													
EFT111	8-224	8-227	56-5#												
EFT114	56-7#														
EFT2	8-426	8-429	8-513	8-516	56-2#										
EFT223	8-447	8-528	56-8#												
EFT336	8-351	8-681	8-687	8-690	56-10#										
EFT337	8-684	56-11#													
EFT344	8-699	56-12#													
EFT353	8-720	56-14#													
EH1	54-1	58-1#													
EH110	54-4	58-2#													
EH111	54-5	58-3#													
EH114	54-7	58-5#													
EH223	54-8	58-6#													
EH256	54-9	58-7#													
EH336	54-11	58-10#													
EH337	54-12	58-11#													
EH344	54-13	58-15#													
EH353	54-15	58-18#													
EHT1	8-3	8-6	8-9	8-12	8-15	8-18	8-24	8-27	8-30	8-33	8-36	8-39	8-42	8-45	
	8-48	8-51	8-54	8-57	8-60	8-63	8-66	8-69	8-72	8-75	8-78	8-81	8-84	8-87	
	8-90	8-93	8-96	8-99	8-102	8-105	8-108	8-111	8-114	8-117	8-120	8-123	8-126	8-130	
	8-133	8-136	8-139	8-142	8-146	8-150	8-154	8-157	8-160	8-163	8-166	8-169	8-172	8-175	
	8-178	8-182	8-185	8-188	8-191	8-194	8-197	8-200	8-203	8-206	8-209	8-212	8-215	8-218	
	8-236	8-239	8-242	8-245	8-248	8-251	8-254	8-257	8-260	8-263	9-266	8-269	8-272	8-275	

	8-278	8-281	8-284	8-287	8-290	8-293	8-296	8-299	8-302	8-305	8-308	8-311	8-314	8-317
	8-320	8-323	8-326	8-329	8-332	8-335	8-338	8-341	8-344	8-347	8-354	8-357	8-360	8-363
	8-366	8-369	8-372	8-375	8-378	8-381	8-384	8-390	8-393	8-396	8-399	8-402	8-405	8-408
	8-411	8-414	8-417	8-420	8-423	8-432	8-435	8-438	8-441	8-444	8-450	8-453	8-456	8-459
	8-504	8-507	8-510	8-519	8-522	8-525	8-531	8-534	8-537	8-540	8-543	8-546	8-549	8-552
	8-556	8-559	8-563	8-566	8-569	8-573	8-577	8-581	8-586	8-590	8-594	8-597	8-600	8-603
	8-606	8-609	8-612	8-615	8-618	8-621	8-624	8-627	8-630	8-633	8-636	8-639	8-642	8-645
	8-648	8-651	8-654	8-657	8-660	8-663	8-666	8-669	8-672	8-675	8-678	8-693	8-696	8-702
	8-705	8-708	8-711	8-714	8-723	54-1#								
EHT110	8-221	54-4#												
EHT111	8-224	8-227	54-5#											
EHT114	54-7#													
EHT2	8-426	8-429	8-513	8-516	54-2#									
EHT223	8-447	54-8#												
EHT256	8-528	54-9#												
EHT336	8-351	8-681	8-687	8-690	54-11#									
EHT337	8-684	54-12#												
EHT344	8-699	54-13#												
EHT353	8-720	54-15#												
EMS1	53-3	57-3#												
EMS10	53-7	53-11	57-10#											
EMS100	53-39	53-71	53-176	53-196	57-66#									
EMS101	53-40	53-59	53-174	53-197	57-67#									
EMS102	53-177	53-178	53-181	53-182	53-183	53-184	53-186	53-187	53-188	53-189	53-234	53-235	57-68#	
EMS103	53-56	53-57	53-135	53-141	53-196	53-197	53-198	53-203	53-204	53-205	53-206	53-207	53-208	53-209
	53-210	53-211	53-213	53-214	53-215	53-219	53-220	57-69#						
EMS104	53-136	53-180	53-203	57-70#										
EMS105	53-204	57-71#												
EMS106	53-212	53-232	57-72#											
EMS11	53-12	53-15	53-16	53-17	53-18	53-19	53-20	53-21	53-22	53-23	53-24	53-25	53-26	53-27
	53-28	53-29	53-30	53-31	53-32	53-33	53-43	53-152	53-177	57-11#				
EMS110	53-9	53-74	53-77	57-73#										
EMS111	53-74	53-226	57-74#											
EMS112	53-74	57-75#												
EMS113	53-75	53-76	53-127	57-77#										
EMS114	53-77	57-78#												
EMS115	53-44	53-45	53-48	53-50	53-51	53-52	53-54	53-55	53-60	53-61	53-63	53-66	53-67	53-70
	53-73	53-77	53-107	53-108	53-109	53-110	53-134	53-137	53-139	53-140	53-142	53-150	53-151	53-153
	53-179	53-182	53-183	53-185	53-186	53-187	53-188	53-189	53-190	53-192	53-193	53-194	53-195	53-201
	53-205	53-221	53-222	53-223	53-232	53-234	57-79#							
EMS116	53-74	53-77	53-191	53-225	57-80#									
EMS117	53-77	53-143	53-148	53-149	53-150	53-178	53-185	53-201	53-225	53-228	53-229	53-230	53-231	57-81#
EMS12	53-12	57-12#												
EMS120	57-82#													
EMS121	53-79	53-88	53-97	53-99	57-83#									
EMS122	53-80	53-69	57-84#											
EMS123	53-81	53-88	53-89	53-90	53-90	53-91	53-92	53-93	53-94	53-95	53-96	53-98	53-126	57-85#
EMS124	53-56	53-82	53-91	53-101	57-86#									
EMS125	53-83	53-92	53-102	53-169	57-87#									
EMS126	53-84	53-93	53-103	57-88#										
EMS127	57-89#													
EMS13	53-13	53-42	53-216	57-13#										
EMS130	53-86	53-95	53-106	57-90#										
EMS131	53-87	53-96	53-104	57-91#										
EMS132	53-79	53-80	53-81	53-82	53-83	53-84	53-85	53-86	53-87	53-88	53-89	53-90	53-91	53-92
	53-93	53-94	53-95	53-96	53-97	53-98	53-99	53-100	53-101	53-102	53-103	53-104	53-105	53-106

CROSS REFERENCE TABLE (CREF V04.00)

	53-125	53-126	53-168	53-169	57-92#										
EMS133	53-79	53-80	53-81	53-82	53-83	53-84	53-85	53-86	53-87	53-125	57-93#				
EMS134	53-88	53-89	53-90	53-91	53-92	53-93	53-94	53-95	53-96	53-126	57-94#				
EMS135	53-97	53-98	57-95#												
EMS136	53-99	53-100	53-101	53-102	53-103	53-104	53-105	53-106	57-96#						
EMS137	53-85	53-94	53-100	57-97#											
EMS14	53-13	57-14#													
EMS140	53-44	53-52	53-60	53-107	53-117	53-137	53-186	53-192	57-98#						
EMS141	53-35	53-44	53-45	53-60	53-61	53-66	53-107	53-122	53-137	53-187	53-188	53-193	53-222	57-99#	
EMS142	53-47	53-65	53-108	53-140	53-184	53-202	57-100#								
EMS143	53-47	53-65	53-73	53-108	53-109	53-110	53-133	53-140	53-184	53-202	53-205	57-101#			
EMS144	53-36	53-73	53-109	53-110	53-205	57-102#									
EMS145	53-36	53-73	53-109	53-110	53-205	57-103#									
EMS146	53-109	53-185	53-190	53-201	57-104#										
EMS147	53-69	53-112	53-113	53-173	53-217	57-105#									
EMS15	53-13	53-42	53-216	57-15#											
EMS150	53-50	53-51	53-54	53-63	53-67	53-111	53-134	53-138	53-182	53-183	53-194	53-195	53-223	53-234	
	57-106#														
EMS151	53-69	53-112	53-113	53-173	53-217	57-107#									
EMS152	53-50	53-51	53-54	53-63	53-111	53-134	53-182	53-183	53-194	53-195	53-223	57-108#			
EMS153	53-111	57-109#													
EMS154	53-111	53-112	53-123	53-124	57-110#										
EMS155	53-112	53-124	57-111#												
EMS156	53-113	53-114	53-115	53-116	53-151	53-173	57-112#								
EMS157	53-113	53-123	57-113#												
EMS16	53-13	57-16#													
EMS160	53-114	53-115	53-116	57-114#											
EMS161	53-114	57-115#													
EMS162	57-116#														
EMS163	53-58	53-59	53-60	53-61	53-62	53-63	53-64	53-65	53-66	53-67	53-68	53-69	53-70	53-71	
	53-72	53-73	53-117	53-143	57-117#										
EMS164	53-45	57-118#													
EMS165	53-56	53-57	53-71	53-72	53-172	53-196	53-197	53-198	57-120#						
EMS166	53-142	57-121#													
EMS167	53-8	53-49	53-68	53-123	53-124	53-168	53-169	53-170	53-175	53-235	57-122#				
EMS17	53-14	57-17#													
EMS170	53-9	57-123#													
EMS171	53-57	53-105	53-125	53-126	57-124#										
EMS172	53-215	57-125#													
EMS173	53-214	57-126#													
EMS174	53-216	57-127#													
EMS175	53-216	53-217	53-218	53-219	57-128#										
EMS176	57-129#														
EMS177	53-127	57-130#													
EMS2	53-4	53-5	57-4#												
EMS20	53-34	53-53	53-58	53-110	53-118	53-119	53-120	53-121	53-212	53-232	53-238	57-18#			
EMS200	53-128	57-131#													
EMS201	53-128	57-132#													
EMS202	53-122	53-130	53-131	53-132	53-133	53-134	53-136	53-180	57-133#						
EMS203	53-122	53-130	53-131	53-132	53-133	53-174	53-175	53-176	53-237	57-134#					
EMS204	53-131	57-135#													
EMS205	53-55	53-142	53-179	53-233	57-136#										
EMS206	53-55	53-142	53-170	53-179	53-233	57-137#									
EMS207	53-168	53-170	57-138#												
EMS21	53-15	57-19#													
EMS210	53-169	53-170	53-173	57-139#											

CROSS REFERENCE TABLE (CREF V04.00)

EMS211	53-170	57-140#												
EMS212	53-171	53-172	57-141#											
EMS213	53-171	57-142#												
EMS214	53-55	57-143#												
EMS215	53-236	57-144#												
EMS216	53-237	57-145#												
EMS217	53-237	57-146#												
EMS22	53-16	57-20#												
EMS220	53-179	57-147#												
EMS221	53-231	53-238	57-148#											
EMS222	53-233	57-149#												
EMS223	53-232	57-150#												
EMS224	53-121	57-151#												
EMS225	53-53	53-150	53-151	53-153	57-152#									
EMS226	53-53	53-150	53-153	57-153#										
EMS227	53-151	57-154#												
EMS23	53-17	57-21#												
EMS230	53-151	53-152	53-153	57-155#										
EMS231	53-151	57-156#												
EMS232	57-157#													
EMS24	53-18	57-22#												
EMS25	53-19	53-115	57-23#											
EMS26	53-20	53-116	53-173	57-24#										
EMS27	53-16	53-19	53-20	53-115	53-116	53-173	57-25#							
EMS3	53-4	53-76	57-5#											
EMS30	53-17	53-18	53-21	53-22	53-23	53-24	53-25	53-26	53-27	53-28	53-29	53-30	53-31	53-32
	53-33	53-43	53-122	53-130	53-131	53-132	53-133	53-144	53-145	53-146	53-147	53-152	53-174	53-175
	53-176	53-177	53-199	57-26#										
EMS31	53-21	53-146	53-207	53-208	57-27#									
EMS32	53-22	57-28#												
EMS33	53-23	53-219	57-29#											
EMS34	53-24	53-225	53-226	53-227	57-30#									
EMS35	53-25	53-216	57-31#											
EMS36	53-26	53-213	53-214	53-215	57-32#									
EMS37	53-27	57-33#												
EMS4	53-5	53-75	57-6#											
EMS40	53-28	53-220	57-34#											
EMS41	53-29	53-190	53-191	57-35#										
EMS42	53-30	53-206	57-36#											
EMS43	53-31	53-209	57-37#											
EMS44	53-32	53-211	53-230	57-38#										
EMS45	53-33	53-210	53-229	57-39#										
EMS46	53-43	53-58	53-120	53-145	53-177	53-185	53-199	53-201	57-40#					
EMS47	53-37	53-44	53-45	53-52	53-60	53-61	53-66	53-117	53-118	53-147	53-186	53-187	53-188	53-192
	53-193	53-222	57-41#											
EMS5	53-6	53-7	57-7#											
EMS50	53-46	53-48	53-64	53-70	53-132	53-135	53-139	53-141	53-181	53-185	53-189	53-200	53-201	53-221
	57-42#													
EMS51	53-48	53-55	53-70	53-130	53-139	53-142	53-179	53-189	53-221	57-43#				
EMS52	53-55	53-142	53-143	53-148	53-149	53-150	53-178	53-179	53-185	53-201	53-228	53-229	53-230	53-231
	53-233	57-44#												
EMS53	53-37	53-38	53-39	53-40	53-44	53-45	53-46	53-47	53-62	53-64	53-65	53-73	53-140	53-141
	53-181	53-182	53-184	53-186	53-187	53-190	53-191	53-192	53-193	53-194	53-195	53-200	53-202	53-203
	53-204	53-206	53-207	53-216	53-218	53-219	53-220	53-226	57-45#					
EMS54	53-226	57-46#												
EMS55	53-227	57-47#												

CROSS REFERENCE TABLE (CREF V04.00)

EMS56	53-50	53-51	53-54	53-62	53-63	53-67	53-119	53-143	53-144	53-179	53-182	53-183	53-194	53-195	SN
	53-223	53-228	53-234	57-48#											SS
EMS57	53-151	53-190	53-191	53-192	53-193	53-194	53-195	53-196	53-197	53-198	53-199	53-200	53-201	53-202	
	53-203	53-204	53-205	53-206	53-207	53-208	53-209	53-210	53-211	53-213	53-214	53-215	53-220	53-221	SS
	53-222	53-223	53-228	53-229	53-230	57-49#									
EMS6	53-6	57-8#													SS
EMS60	53-224	57-50#													SS
EMS61	53-224	57-51#													SS
EMS62	53-224	57-52#													ST
EMS63	53-42	53-149	53-190	57-53#											
EMS64	53-8	53-14	53-41	53-144	53-145	53-146	53-147	53-199	57-54#						ST
EMS65	53-41	57-55#													ST
EMS66	53-34	53-218	57-56#												ST
EMS67	53-43	53-44	53-45	53-46	53-47	53-48	53-49	53-50	53-51	53-52	53-54	57-57#			ST
EMS7	53-10	57-9#													ST
EMS70	53-48	53-50	53-52	53-54	53-66	53-70	53-73	53-107	53-110	53-139	53-182	53-189	53-194	53-205	ST
	53-221	57-58#													ST
EMS71	53-51	53-59	53-63	53-183	53-195	53-208	57-59#								ST
EMS72	53-36	53-48	53-51	53-55	53-56	53-57	53-60	53-61	53-61	53-63	53-108	53-109	53-134	53-135	ST
	53-139	53-140	53-142	53-153	53-179	53-183	53-187	53-193	53-195	53-232	57-60#				ST
EMS73	53-49	53-54	53-67	53-68	53-136	53-180	53-188	53-222	53-223	53-234	53-235	57-61#			ST
EMS74	53-69	53-70	53-134	53-189	53-217	53-221	57-62#								ST
EMS75	53-35	53-66	53-67	53-137	53-138	53-188	53-222	53-223	53-234	57-63#					ST
EMS76	53-35	53-36	53-37	53-38	53-39	53-40	57-64#								
EMS77	53-38	53-72	53-148	53-198	57-65#										ST
EMT1	8-3	53-3#													ST
EMT10	8-24	53-10#													ST
EMT100	8-197	53-66#													ST
EMT101	8-200	53-67#													SW
EMT102	8-203	53-68#													SW
EMT103	8-206	53-69#													SW
EMT104	8-209	53-70#													SW
EMT105	8-212	53-71#													SW
EMT106	8-215	53-72#													SW
EMT107	8-218	53-73#													SW
EMT11	8-27	53-11#													SW
EMT110	8-221	53-74#													SW
EMT111	8-224	53-75#													SW
EMT112	8-227	53-76#													SW
EMT113	8-230	53-77#													SW
EMT114	53-78#														SW
EMT115	8-236	53-79#													SW
EMT116	8-239	53-80#													SW
EMT117	8-242	53-81#													SW
EMT12	8-30	53-12#													SW
EMT120	8-245	53-82#													SW
EMT121	8-248	53-83#													SW
EMT122	8-251	53-84#													SW
EMT123	8-254	53-85#													SW
EMT124	8-257	53-86#													SW
EMT125	8-260	53-87#													SW
EMT126	8-263	53-88#													SW
EMT127	8-266	53-89#													SW
EMT13	8-33	53-13#													SW
EMT130	8-269	53-90#													SW
EMT131	8-272	53-91#													SW

CROSS REFERENCE TABLE (CREF V04.00)

EMT132	8-275	53-92#
EMT133	8-278	53-93#
EMT134	8-281	53-94#
EMT135	8-284	53-95#
EMT136	8-287	53-96#
EMT137	8-290	53-97#
EMT14	8-36	53-14#
EMT140	8-293	53-98#
EMT141	8-296	53-99#
EMT142	8-299	53-100#
EMT143	8-302	53-101#
EMT144	8-305	53-102#
EMT145	8-308	53-103#
EMT146	8-311	53-104#
EMT147	8-314	53-105#
EMT15	8-39	53-15#
EMT150	8-317	53-106#
EMT151	8-320	53-107#
EMT152	8-323	53-108#
EMT153	8-326	53-109#
EMT154	8-329	53-110#
EMT155	8-332	53-111#
EMT156	8-335	53-112#
EMT157	8-338	53-113#
EMT16	8-42	53-16#
EMT160	8-341	53-114#
EMT161	8-344	53-115#
EMT162	8-347	53-116#
EMT163	53-117#	
EMT164	8-354	53-118#
EMT165	8-357	53-119#
EMT166	8-360	53-120#
EMT167	8-363	53-121#
EMT17	8-45	53-17#
EMT170	8-366	53-122#
EMT171	8-369	53-123#
EMT172	8-372	53-124#
EMT173	8-375	53-125#
EMT174	8-378	53-126#
EMT175	8-381	53-127#
EMT176	8-384	53-128#
EMT177	53-129#	
EMT2	8-6	53-4#
EMT20	8-48	53-18#
EMT200	8-390	53-130#
EMT201	8-393	53-131#
EMT202	8-396	53-132#
EMT203	8-399	53-133#
EMT204	8-402	53-134#
EMT205	8-405	53-135#
EMT206	8-408	53-136#
EMT207	8-411	53-137#
EMT21	8-51	53-19#
EMT210	8-414	53-138#
EMT211	8-417	53-139#
EMT212	8-420	53-140#

CROSS REFERENCE TABLE (CREF V04.00)

EMT213	8-423	53-141#
EMT214	8-426	53-142#
EMT215	8-429	53-143#
EMT216	8-432	53-144#
EMT217	8-435	53-145#
EMT22	8-54	53-20#
EMT220	8-438	53-146#
EMT221	8-441	53-147#
EMT222	8-444	53-148#
EMT223	8-447	53-149#
EMT224	8-450	53-150#
EMT225	8-453	53-151#
EMT226	8-456	53-152#
EMT227	8-459	53-153#
EMT23	8-57	53-21#
EMT230	8-462	53-154#
EMT231	8-465	53-155#
EMT232	8-468	53-156#
EMT233	8-471	53-157#
EMT234	53-158#	
EMT235	53-159#	
EMT236	53-160#	
EMT237	53-161#	
EMT24	8-60	53-22#
EMT240	53-162#	
EMT241	53-163#	
EMT242	53-164#	
EMT243	53-165#	
EMT244	53-166#	
EMT245	53-167#	
EMT246	8-504	53-168#
EMT247	8-507	53-169#
EMT25	8-63	53-23#
EMT250	8-510	53-170#
EMT251	8-513	53-171#
EMT252	8-516	53-172#
EMT253	8-519	53-173#
EMT254	8-522	53-174#
EMT255	8-525	53-175#
EMT256	8-528	53-176#
EMT257	8-531	53-177#
EMT26	8-66	53-24#
EMT260	8-534	53-178#
EMT261	8-537	53-179#
EMT262	8-540	53-180#
EMT263	8-543	53-181#
EMT264	8-546	53-182#
EMT265	8-549	53-183#
EMT266	8-552	53-184#
EMT267	8-556	53-185#
EMT27	8-69	53-25#
EMT270	8-559	53-186#
EMT271	8-563	53-187#
EMT272	8-566	53-188#
EMT273	8-569	53-189#
EMT274	8-573	53-190#

TY
TY
TY
U0
U1
U2
UN
UN
UN
UN
UP
US
US
VV

WC
WC
WC
WC
WC
WC
WC
WD

WH

WL
WR
XS
XX
Y
ZE

EMT275 8-577 53-191#
EMT276 8-581 53-192#
EMT277 8-586 53-193#
EMT3 8-9 53-5#
EMT30 8-72 53-26#
EMT300 8-590 53-194#
EMT301 8-594 53-195#
EMT302 8-597 53-196#
EMT303 8-600 53-197#
EMT304 8-603 53-198#
EMT305 8-606 53-199#
EMT306 8-609 53-200#
EMT307 8-612 53-201#
EMT31 8-75 53-27#
EMT310 8-615 53-202#
EMT311 8-618 53-203#
EMT312 8-621 53-204#
EMT313 8-624 53-205#
EMT314 8-627 53-206#
EMT315 8-630 53-207#
EMT316 8-633 53-208#
EMT317 8-636 53-209#
EMT32 8-78 53-28#
EMT320 8-639 53-210#
EMT321 8-642 53-211#
EMT322 8-645 53-212#
EMT323 8-648 53-213#
EMT324 8-651 53-214#
EMT325 8-654 53-215#
EMT326 8-657 53-216#
EMT327 8-660 53-217#
EMT33 8-81 53-29#
EMT330 8-663 53-218#
EMT331 8-666 53-219#
EMT332 8-669 53-220#
EMT333 8-672 53-221#
EMT334 8-675 53-222#
EMT335 8-678 53-223#
EMT336 8-351 8-681 53-224#
EMT337 8-684 53-225#
EMT34 8-84 53-30#
EMT340 8-687 53-226#
EMT341 8-690 53-227#
EMT342 8-693 53-228#
EMT343 8-696 53-229#
EMT344 8-699 53-230#
EMT345 8-702 53-231#
EMT346 8-705 53-232#
EMT347 8-708 53-233#
EMT35 8-87 53-31#
EMT350 8-711 53-234#
EMT351 8-714 53-235#
EMT352 8-717 53-236#
EMT353 8-720 53-237#
EMT354 8-723 53-238#
EMT36 8-90 53-32#

CROSS REFERENCE TABLE (CREF V04.00)

EMT37	8-93	53-33#												
EMT4	8-12	53-6#												
EMT40	8-96	53-34#												
EMT41	8-99	53-35#												
EMT42	8-102	53-36#												
EMT43	8-105	53-37#												
EMT44	8-108	53-38#												
EMT45	8-111	53-39#												
EMT46	8-114	53-40#												
EMT47	8-117	53-41#												
EMT5	8-15	53-7#												
EMT50	8-120	53-42#												
EMT51	8-123	53-43#												
EMT52	8-126	53-44#												
EMT53	8-130	53-45#												
EMT54	8-133	53-46#												
EMT55	8-136	53-47#												
EMT56	8-139	53-48#												
EMT57	8-142	53-49#												
EMT6	8-18	53-8#												
EMT60	8-146	53-50#												
EMT61	8-150	53-51#												
EMT62	8-154	53-52#												
EMT63	8-157	53-53#												
EMT64	8-160	53-54#												
EMT65	8-163	53-55#												
EMT66	8-166	53-56#												
EMT67	8-169	53-57#												
EMT7	8-21	53-9#												
EMT70	8-172	53-58#												
EMT71	8-175	53-59#												
EMT72	8-178	53-60#												
EMT73	8-182	53-61#												
EMT74	8-185	53-62#												
EMT75	8-188	53-63#												
EMT76	8-191	53-64#												
EMT77	8-194	53-65#												
EMTVEC	4-492#	10-23*	10-23*											
ENRGDT	52-144#													
EQUALS	49-4#													
ERR	4-560#	25-80	25-82	25-98	25-132									
ERRNMB	43-40*	43-41*	43-44	43-49	43-56	43-152#								
ERROR	4-492#													
ERRTYP	42-1	43-13#												
ERRVEC	4-492#	10-23	10-23*	10-23*	10-25*	10-26*	13-5	13-5	13-6*	13-7*	13-20*	13-20*	13-24*	13-24*
	20-35	20-35	20-39*	20-40*	20-70*	20-70*	21-29	21-29	21-33*	21-34*	21-63*	21-63*	22-3	22-3
	22-4*	22-5*	22-11*	22-18*	22-18*	23-9	23-9	23-10*	23-11*	23-31*	23-31*	27-20	27-20	27-21*
	27-22*	27-51*	27-51*	29-12	29-12	29-13*	29-14*	29-25*	29-25*	41-1	41-1	41-1*	41-1*	41-1*
	41-1*	41-1*	42-1	42-1*	42-1*									
ERTY00	43-21	43-157#												
ERTY01	43-38	43-158#												
ERTY02	43-43	43-159#												
ERTY03	43-45	43-160#												
ERTY04	43-144	43-161#												
ERTY05	43-31	43-162#												
ESRC	4-632#													

CROSS REFERENCE TABLE (CREF V04.00)

F0	4-503#	25-28												
F1	4-502#	25-28												
F2	4-501#	25-28	25-675											
F3	4-500#	25-28												
F4	4-499#	25-28												
FER	4-584#	4-590	13-A40	13-A45	13-B48	13-B53	13-C92	13-C95	13-D58	13-D61	16-221	25-412	34-352	34-369
	34-372	34-395	34-398											
FIND	13-41	13-41#	13-41#	13-52	13-52#	13-52#	13-68	13-68#	13-68#	13-71	13-71#	13-73	13-73#	13-76
	13-76#	13-76#	13-80	13-80#	13-80#	13-93	13-93#	13-93#	13-110	13-110#	13-110#	13-113	13-113#	13-115
	13-115#	13-118	13-118#	13-118#	13-122	13-122#	13-122#	13-124	13-124#	13-124#	13-126	13-126#	13-126#	13-139
	13-139#	13-139#	13-155	13-155#	13-155#	13-158	13-158#	13-160	13-160#	13-163	13-163#	13-163#	13-167	13-167#
	13-167#	13-169	13-169#	13-169#	13-171	13-171#	13-171#	13-175	13-175#	13-175#	13-188	13-188#	13-188#	13-199
	13-199#	13-199#	13-215	13-215#	13-215#	13-218	13-218#	13-220	13-220#	13-223	13-223#	13-223#	13-227	13-227#
	13-227#	13-240	13-240#	13-240#	13-257	13-257#	13-257#	13-260	13-260#	13-262	13-262#	13-265	13-265#	13-265#
	13-269	13-269#	13-269#	13-271	13-271#	13-271#	13-273	13-273#	13-273#	13-286	13-286#	13-286#	13-301	13-301#
	13-301#	13-304	13-304#	13-306	13-306#	13-309	13-309#	13-309#	13-313	13-313#	13-313#	13-315	13-315#	13-315#
	13-317	13-317#	13-317#	13-330	13-330#	13-330#	13-341	13-341#	13-341#	13-356	13-356#	13-356#	13-359	13-359#
	13-361	13-361#	13-364	13-364#	13-364#	13-368	13-368#	13-368#	13-381	13-381#	13-381#	13-398	13-398#	13-398#
	13-401	13-401#	13-403	13-403#	13-406	13-406#	13-406#	13-410	13-410#	13-410#	13-412	13-412#	13-412#	13-414
	13-414#	13-414#	13-428	13-428#	13-428#	13-443	13-443#	13-443#	13-446	13-446#	13-448	13-448#	13-451	13-451#
	13-451#	13-455	13-455#	13-455#	13-460	13-460#	13-460#	13-488	13-488#	13-488#	13-504	13-504#	13-504#	13-515
	13-515#	13-515#	13-533	13-533#	13-533#	13-536	13-536#	13-538	13-538#	13-541	13-541#	13-541#	13-545	13-545#
	13-545#	13-559	13-559#	13-559#	13-576	13-576#	13-576#	13-579	13-579#	13-581	13-581#	13-584	13-584#	13-584#
	13-588	13-588#	13-588#	13-590	13-590#	13-590#	13-592	13-592#	13-592#	13-605	13-605#	13-605#	13-621	13-621#
	13-621#	13-624	13-624#	13-626	13-626#	13-629	13-629#	13-629#	13-633	13-633#	13-633#	13-635	13-635#	13-635#
	13-637	13-637#	13-637#	13-639	13-639#	13-639#	13-652	13-652#	13-652#	13-663	13-663#	13-663#	13-681	13-681#
	13-681#	13-684	13-684#	13-686	13-686#	13-689	13-689#	13-689#	13-693	13-693#	13-693#	13-707	13-707#	13-707#
	13-724	13-724#	13-724#	13-727	13-727#	13-729	13-729#	13-732	13-732#	13-732#	13-736	13-736#	13-736#	13-738
	13-738#	13-738#	13-740	13-740#	13-740#	13-753	13-753#	13-753#	13-768	13-768#	13-768#	13-771	13-771#	13-773
	13-773#	13-776	13-776#	13-776#	13-780	13-780#	13-780#	13-782	13-782#	13-782#	13-784	13-784#	13-784#	13-797
	13-797#	13-797#	13-808	13-808#	13-808#	13-826	13-826#	13-826#	13-829	13-829#	13-831	13-831#	13-834	13-834#
	13-834#	13-838	13-838#	13-838#	13-852	13-852#	13-852#	13-869	13-869#	13-869#	13-872	13-872#	13-874	13-874#
	13-877	13-877#	13-877#	13-881	13-881#	13-881#	13-883	13-883#	13-883#	13-885	13-885#	13-885#	13-899	13-899#
	13-899#	13-914	13-914#	13-914#	13-917	13-917#	13-919	13-919#	13-922	13-922#	13-922#	13-926	13-926#	13-926#
	13-930	13-930#	13-930#	13-960	13-960#	13-960#	13-985	13-985#	13-985#	13-988	13-988#	13-988#	13-:06	13-:06#
	13-:06#	13-:09	13-:09#	13-:11	13-:11#	13-:14	13-:14#	13-:14#	13-:18	13-:18#	13-:18#	13-:40	13-:40#	13-:40#
	13-:55	13-:55#	13-:55#	13-:58	13-:58#	13-:60	13-:60#	13-:63	13-:63#	13-:63#	13-:67	13-:67#	13-:67#	13-:69
	13-:69#	13-:69#	13-:71	13-:71#	13-:71#	13-:84	13-:84#	13-:84#	13-:99	13-:99#	13-:99#	13-;02	13-;02#	13-;04
	13-;04#	13-;07	13-;07#	13-;07#	13-;11	13-;11#	13-;11#	13-;13	13-;13#	13-;13#	13-;15	13-;15#	13-;15#	13-;31
	13-;31#	13-;31#	13-;43	13-;43#	13-;43#	13-;61	13-;61#	13-;61#	13-;64	13-;64#	13-;66	13-;66#	13-;69	13-;69#
	13-;69#	13-;73	13-;73#	13-;73#	13-;86	13-;86#	13-;86#	13-;91	13-;91#	13-;91#	13-;94	13-;94#	13-;96	13-;96#
	13-;99	13-;99#	13-;99#	13-<03	13-<03#	13-<03#	13-<21	13-<21#	13-<21#	13-<24	13-<24#	13-<27	13-<27#	13-<27#
	13-<31	13-<31#	13-<31#	13-<33	13-<33#	13-<33#	13-<35	13-<35#	13-<35#	13-<48	13-<48#	13-<48#	13-<53	13-<53#
	13-<53#	13-<56	13-<56#	13-<58	13-<58#	13-<61	13-<61#	13-<61#	13-<65	13-<65#	13-<65#	13-<83	13-<83#	13-<83#
	13-<86	13-<86#	13-<89	13-<89#	13-<89#	13-<93	13-<93#	13-<93#	13-<95	13-<95#	13-<95#	13-<97	13-<97#	13-<97#
	13-<99	13-<99#	13-<99#	13-=33	13-=33#	13-=33#	13-=48	13-=48#	13-=48#	13-=51	13-=51#	13-=53	13-=53#	13-=56
	13-=56#	13-=56#	13-=60	13-=60#	13-=60#	13-=78	13-=78#	13-=78#	13-=81	13-=81#	13-=83	13-=83#	13-=86	13-=86#
	13-=86#	13-=95	13-=95#	13-=95#	13-=97	13-=97#	13-=97#	13-=99	13-=99#	13-=99#	13->12	13->12#	13->12#	13->16
	13->16#	13->16#	13->19	13->19#	13->21	13->21#	13->24	13->24#	13->24#	13->28	13->28#	13->28#	13->43	13->43#
	13->43#	13->46	13->46#	13->48	13->48#	13->51	13->51#	13->51#	13->60	13->60#	13->60#	13->62	13->62#	13->62#
	13->64	13->64#	13->64#	13->98	13->98#	13->98#	13-?16	13-?16#	13-?16#	13-?19	13-?19#	13-?21	13-?21#	13-?24
	13-?24#	13-?24#	13-?33	13-?33#	13-?33#	13-?35	13-?35#	13-?35#	13-?37	13-?37#	13-?37#	13-?50	13-?50#	13-?50#
	13-?65	13-?65#	13-?65#	13-?68	13-?68#	13-?70	13-?70#	13-?73	13-?73#	13-?73#	13-?82	13-?82#	13-?82#	13-?84
	13-?84#	13-?84#	13-?86	13-?86#	13-?86#	13-a06	13-a06#	13-a06#	13-a17	13-a17#	13-a17#	13-a41	13-a41#	13-a41#
	13-a44	13-a44#	13-a46	13-a46#	13-a49	13-a49#	13-a49#	13-a53	13-a53#	13-a53#	13-a66	13-a66#	13-a66#	13-a83
	13-a83#	13-a83#	13-a86	13-a86#	13-a88	13-a88#	13-a91	13-a91#	13-a91#	13-a95	13-a95#	13-a95#	13-A14	13-A14#

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

	13-F53	13-G08	13-G64	13-H14	13-H58	13-I10	13-I62	13-J07	13-J59	13-K15	13-K25	13-K75	13-K86	13-L44
	15-125	15-183	16-86	16-118	16-152	16-181	23-17	24-90	24-102	24-107	24-123	25-47	25-50	34-346
GTSWR	10-28	46-1#												
HCE	4-581#	4-590	13-A11	13-A17	13-B19	13-B25	13-C92	13-C95	13-D58	13-D61	16-221	25-375	25-380	25-395
	25-397	25-410	25-425	25-451	25-456	34-352	34-382	34-385	34-395	34-398	50-77	50-78	50-81	50-85
	50-86													
HCI	4-659#	13-C64	25-373	34-348										
HCRC	4-580#	4-590	16-221	25-382	34-352	34-356	34-359	34-395	34-398					
HELP	11-17	61-23#												
HT	4-492#	40-1	40-1	43-82										
IAE	4-578#	25-279	25-651	25-667	25-694	28-48	28-72	32-24	32-78	32-81	34-172	34-194	50-59	50-69
	50-77	50-78	50-81	50-82	50-85	50-86								
IBSAVE	42-1	42-1	42-1	42-1	42-1	42-1#	42-1*	42-1*	42-1*					
IDXMSK	4-720#	20-51	20-54	21-45	21-48									
IE	4-733#	4-766#												
ILF	4-588#	25-151	25-153	31-51	31-111	31-115	32-24	32-45	32-48	34-128	34-145	34-148	50-58	50-67
	50-68	50-70	50-71	50-72	50-73	50-74	50-75	50-76	50-79	50-80	50-83	50-84	50-87	50-88
ILF02	4-509#													
ILF24	4-519#													
ILF26	4-520#													
ILF30	4-524#													
ILF32	4-524#													
ILF34	4-524#													
ILF36	4-524#													
ILF40	4-524#													
ILF42	4-524#													
ILF44	4-524#													
ILF46	4-524#													
ILF54	4-527#													
ILF56	4-528#													
ILF64	4-531#													
ILF66	4-532#													
ILF74	4-535#													
ILF76	4-536#													
ILR	4-587#	31-51	31-97	31-101	32-232	32-236	32-239	34-128	34-131	34-134				
ILRG50	4-718#													
ILRG52	4-718#													
ILRG54	4-718#													
ILRG56	4-718#													
ILRG60	4-718#													
ILRG62	4-718#													
ILRG64	4-718#													
ILRG66	4-718#													
ILRG70	4-718#													
ILRG72	4-718#													
ILRG74	4-718#													
ILRG76	4-718#													
IOTVEC	4-492#	10-23*	10-23*											
IPCK0	4-770#													
IPCK1	4-769#													
IPCK2	4-768#													
IPCK3	4-767#													
IR	4-746#	30-56												
IVC	4-685#	13-E53	13-E60	13-F13	13-F19	25-194	25-195	28-90	28-213	32-92	32-99	32-102	32-171	34-111
	34-115	34-550	35-58	50-58	50-59	50-60	50-62	50-63	50-64	50-67	50-68	50-69	50-70	50-71
	50-72	50-73	50-74	50-75	50-76	50-77	50-78	50-79	50-80	50-81	50-82	50-83	50-84	50-85

CROSS REFERENCE TABLE (CREF V04.00)

```
        50-86    50-87    50-88
LBC      4-687#  34-236   34-254   34-259
LBT      4-564#  25-354   25-653   25-655
LF       4-492#  40-1     40-1     43-80    49-18    57-79
LODEV   10-117   49-28#
LS       4-641#  30-80    33-70
LSC      4-686#  34-236   34-270   34-273
LST      4-642#  30-80    33-70
LSTRK    7-0#    12-57*   13->80   13-?90   13-J38   13-L77   16-273   16-384   16-410   17-54    25-590   28-54    34-178
MCLK     4-615#
MCPE     4-728#  24-90    24-107   24-120   24-123   34-27    34-30
MDF      4-620#
MDPE     4-744#  25-481   34-442   34-445
MEDENB   7-0#    12-48*   16-39    16-291*
MFGFIL  16-246   16-275   16-330   16-475*  61-7#
MI       4-623#
MIXED   13-114   52-4#
MOC      4-618#
MOH      4-649#
MOL      4-562#  10-98    28-151   28-161   28-162   28-167   28-172   28-200   31-37    31-40    32-68    32-145   32-146   32-153
        32-158   33-30    34-88    34-256   34-509   34-510   34-532   34-537   35-35    35-36    35-40    35-45
MRD      4-616#
MS       4-621#
MSC      4-624#
MSDRVS  11-64    49-23#
MSE      4-694#  13-A06   13-B14   17-29
MSER     4-619#
MSGDRV  12-61    49-24#
MSHELP  11-11    49-8#
MUR      4-617#
MWD      4-640#  30-81    33-71
MWP      4-622#
MXF      4-743#  25-339   34-50    34-63    34-66
N       11-19    49-35#
NDTMSK   4-590#
NED      4-740#  10-87    24-72    24-76    27-31
NEM      4-741#  25-324
NONE    12-7     49-34#
NOP      4-508#
NOTAVL  10-105   49-30#
NOTPRS  10-102   49-29#
NOTRM   10-96    49-27#
NSA      4-647#
OCC      4-628#
OFD      4-661#  13-L09   13-L11
OFFSET   4-514#  13-K15   13-K75
OM       4-569#  32-145   32-201   32-204   33-29    34-457   34-460
ONES    13-517   13-665   13-810   13-:28   13-;45   13-C10   52-21#
OPE      4-684#
OPI      4-575#  25-178   28-145   28-169   31-51    31-69    31-73    32-24    32-61    32-64    32-155   34-83    34-86    34-534
        35-42    50-57    50-58    50-59    50-60    50-61    50-62    50-63    50-64    50-65    50-66    50-67    50-68    50-69
        50-70    50-71    50-72    50-73    50-74    50-75    50-76    50-77    50-78    50-79    50-80    50-81    50-82    50-83
        50-84    50-85    50-86    50-87    50-88
OR       4-745#
PACACK   4-518#  16-118
PAKACK   4-517#   4-518   15-125
```

CROSS REFERENCE TABLE (CREF V04.00)

```
PAR      4-585#  24-135  24-139  24-151  28-30   32-24   32-29   32-34   34-41   34-46
PAT      4-748#
PDA      4-635#
PFECH   43-59    43-164#
PFECH1  43-164   43-165#
PFECH2  43-164   43-168#
PFECH3  43-164   43-171#
PFECH4  43-164   43-173#
PGE      4-742#
PGM      4-565#  33-29
PHA      4-636#
PIP      4-561#  13-F71  13-F74  13-G26  13-G29  15-180  16-149  28-161  28-182  28-187  32-145  32-216  32-221  33-29
        34-509   34-515  34-520  35-35   35-72   35-77
PIRQ     4-492#
PIRQVE   4-492#
PLFS     4-633#
PR0      4-492#
PR1      4-492#
PR2      4-492#
PR3      4-492#  41-5
PR4      4-492#
PR5      4-492#  12-50   41-9
PR6      4-492#  10-26   13-7    20-40   21-34   22-5    23-11   27-22   29-14
PR7      4-492#
PRIERR  13-122   13-167  13-269  13-313  13-410  13-455  13-588  13-633  13-736  13-780  13-881  13-926  13-:67  13-:11
        13-<31   13-<93  13-=95  13->60  13-?33  13-?82  13-@95  13-B05  13-C89  13-D55  13-E51  13-F11  13-F84  13-G40
        13-H36   13-H81  13-I84  13-J29  13-K42  13-L03  13-L65  24-39#
PS       4-492    4-492#
PSEL     4-729#
PSW      4-492#
PUT     13-68    13-110  13-155  13-215  13-257  13-301  13-356  13-398  13-443  13-533  13-576  13-621  13-681  13-724
        13-768   13-826  13-869  13-914  13-:06  13-:55  13-:99  13-;61  13-;91  13-<21  13-<53  13-<83  13-=48  13-=78
        13->16   13->43  13-?16  13-?65  13-@41  13-@83  13-A93  13-C34  13-C77  13-D43  13-E22  13-E39  13-E81  13-E99
        13-F48   13-F63  13-G03  13-G18  13-G82  13-H24  13-H69  13-I30  13-I72  13-J17  13-J77  13-K14  13-K20  13-K30
        13-K71   13-K80  13-K91  13-L54  15-128  15-186  16-87   16-119  16-153  16-182  21-28#
PUTBUF   7-0#    16-48   16-292* 21-31
PUTINX   7-0#    13-59   13-101  13-146  13-206  13-248  13-292  13-347  13-389  13-434  13-522  13-567  13-612  13-670
        13-715   13-759  13-815  13-860  13-905  13-995  13-:46  13-:90  13-;50  13-<12  13-<74  13-=38  13-=69  13->34
        13-?07   13-?56  13-@32  13-@74  13-A84  13-C23  13-C68  13-D34  13-E22* 13-E22* 13-E29  13-E81* 13-E81* 13-E89
        13-F48*  13-F48* 13-F54  13-G03* 13-G03* 13-G09  13-G73  13-H15  13-H60  13-I19  13-I63  13-J08  13-J68  13-K06
        13-K16   13-K26  13-K63  13-K76  13-K87  13-L45  15-126* 15-127* 15-184* 15-185* 16-62   21-32   24-142
PWRVEC   4-492#  10-23*  10-23*  47-1*   47-1*   47-1*   47-1*   47-1*   47-1*
QUES     49-6#
R6       4-492#  10-23   10-23*  10-23*
R7       4-492#
RCLSTS  15-209   16-170  32-15#
RD       4-533#  13-145  13-611  13-<72  13-=22  13-=65  13->87  13-?03  13-A83  13-D32  13-E86  13-G08  13-H58  13-J07
        13-K25
RDCHR   11-12    11-65   11-82   44-1    46-1#
RDLIN   45-1     46-1#
RDOCT   11-35    11-49   46-1#
RDY      4-732#  23-17   23-18   24-86   24-89   30-31
READY   12-37#   14-24
RECAL    4-511#  15-183  16-152
RESREG  43-148   46-1#
RESVEC   4-492#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
REX       4-631#
RG        4-629#
RGDTPT    52-3#
RH        4-534#  16-181
RIP       4-516#
RLEASE    4-513#
RMAS      4-704#
RMASI     7-0#    33-50   59-13
RMASO     7-0#
RMBA      4-776#  13-12*  13-13   13-62   13-105  13-150  13-209  13-252  13-296  13-350  13-393  13-438  13-525  13-571
          13-616  13-673  13-719  13-763  13-818  13-864  13-909  13-998  13-:50  13-:94  13-;53  13-<16  13-<78  13-=41
          13-=73  13->38  13-?11  13-?60  13-@35  13-@78  13-A88  13-C26  13-C72  13-D38  13-E35  13-E94  13-F58  13-G13
          13-G76  13-H19  13-H64  13-I22  13-I67  13-J12  13-J71  13-K10  13-K67  13-L50  16-67
RMBAE     4-779#
RMBAEI    7-0#
RMBAEO    7-0#
RMBAI     7-0#    13-476  13-948  25-560  25-562  30-42   59-14
RMBAO     7-0#    13-47*  13-98*  13-144* 13-194* 13-245* 13-336* 13-387* 13-510* 13-564* 13-610* 13-658* 13-712* 13-803*
          13-858* 13-980* 13-:26* 13-;38* 13-<10* 13-<73* 13-=20* 13->86* 13-@12* 13-@71* 13-A81* 13-C03* 13-C67* 13-D33*
          13-E06* 13-E87* 13-F37* 13-G61* 13-H13* 13-H59* 13-I07* 13-I60* 13-J06* 13-J56* 13-J97* 13-L23* 16-60*  16-246
          16-275* 16-279  16-282* 17-21   18-31   25-554  25-558
RMCS1     4-700#  4-774#  10-89   13-?*   13-65   13-107  13-152  13-212  13-254  13-298  13-353  13-395  13-440  13-528
          13-573  13-618  13-676  13-721  13-765  13-821  13-866  13-911  13-:01  13-:52  13-:96  13-;56  13-<18  13-<80
          13-=44  13-=75  13->40  13-?13  13-?62  13-@38  13-@80  13-A90  13-C29  13-C74  13-D40  13-E36  13-E96  13-F60
          13-G15  13-G79  13-H21  13-H66  13-I25  13-I69  13-J14  13-J74  13-K17  13-K27  13-K77  13-K88  13-L51  15-126
          15-184  16-68   20-42   21-36   23-16   27-28
RMCS1I    7-0#    16-208  24-65   24-67   24-69   24-86   24-88   24-91   24-102  24-106  24-108  24-120  24-122  24-124
          25-47   25-49   25-60   25-103  25-540  30-29   33-16   34-27   34-29   34-31   59-13
RMCS1O    7-0#    13-50*  13-100* 13-145* 13-197* 13-247* 13-291* 13-339* 13-388* 13-433* 13-513* 13-566* 13-611* 13-661*
          13-714* 13-758* 13-806* 13-859* 13-904* 13-982* 13-:27* 13-:45* 13-:89* 13-;41* 13-;89* 13-<11* 13-<51* 13-<72*
          13-=22* 13-=37* 13-=65* 13->14* 13->33* 13->87* 13-?03* 13-?55* 13-@15* 13-@73* 13-A83* 13-C06* 13-C66* 13-D32*
          13-E09* 13-E27* 13-E86* 13-F40* 13-F53* 13-G08* 13-G64* 13-H14* 13-H58* 13-I10* 13-I62* 13-J07* 13-J59* 13-K15*
          13-K25* 13-K75* 13-K86* 13-L24* 13-L44* 15-125* 15-183* 16-86*  16-118* 16-152* 16-181* 16-312  17-26   25-27
          25-576  34-324  34-344
RMCS2     4-777#  10-84*  10-85*  10-87   12-55*  12-56*  13-14   13-15*  13-16   13-17*  20-41   21-35   27-27*  27-29
          29-16*  29-18*
RMCS2I    7-0#    13-457  13-462  13-464  13-928  13-932  13-934  24-47   24-72   24-74   24-75   25-95   25-166  25-267
          25-310  25-325  25-340  25-482  25-556  30-51   34-50   34-63   34-65   34-67   34-442  34-444  34-446  34-496
          34-498  34-500  59-13
RMCS2O    7-0#
RMCS3     4-780#
RMCS3I    7-0#
RMCS3O    7-0#
RMDA      4-701#  13-61   13-102  13-147  13-208  13-249  13-293  13-349  13-390  13-435  13-524  13-568  13-613  13-672
          13-716  13-760  13-817  13-861  13-906  13-997  13-:47  13-:91  13-;52  13-<13  13-<75  13-=40  13-=70  13->35
          13-?08  13-?57  13-@34  13-@75  13-A85  13-C25  13-C69  13-D35  13-E31  13-E91  13-F55  13-G10  13-G75  13-H16
          13-H61  13-I21  13-I64  13-J09  13-J70  13-K07  13-K64  13-L46  16-63
RMDAI     7-0#    25-628  25-699  59-14   59-16
RMDAO     7-0#    13-46*  13-193* 13-335* 13-509* 13-657* 13-802* 13-979* 13-:37* 13-=16* 13->68* 13->80* 13->81* 13-?90*
          13-?91* 13-@11* 13-C02* 13-E05* 13-E13  13-F36* 13-G60* 13-I05* 13-J37* 13-J38  13-J55* 13-L19* 13-L71* 13-L72
          13-L75* 13-L76* 13-L77  13-L80* 16-57*  16-233* 16-238* 16-241* 16-242  16-248  16-250  16-273* 16-281* 16-307
          16-425* 17-24   25-588  25-589  25-628  28-54   28-57   28-61   28-65   34-178  34-181  34-185  34-189
RMDB      4-778#  13-18*  13-19   13-470  13-941  19-20
RMDBI     7-0#    13-472  13-941* 13-944
RMDBO     7-0#
RMDC      4-710#  13-60   13-103  13-149  13-207  13-250  13-295  13-348  13-391  13-437  13-523  13-569  13-615  13-671
```

CROSS REFERENCE TABLE (CREF V04.00)

	13-717	13-762	13-816	13-862	13-908	13-996	13-:48	13-:93	13-;51	13-<14	13-<77	13-=39	13-=71	13->37
	13-?09	13-?59	13-@33	13-@76	13-A87	13-C24	13-C70	13-D37	13-E32	13-E93	13-F56	13-G12	13-G74	13-H17
	13-H63	13-I20	13-I65	13-J11	13-J69	13-K08	13-K66	13-L47	16-64					
RMDCI	7-0#	25-712	59-14	59-16										
RMDCO	7-0#	13-45*	13-192*	13-334*	13-508*	13-656*	13-801*	13-978*	13-;35*	13-;36	13-;88*	13-<08*	13-<50*	13-<71*
	13-=19*	13->84*	13-@10*	13-C01*	13-E04*	13-F35*	13-G59*	13-I04*	13-J54*	13-L18*	13-L28	13-L81*	13-L82	16-56*
	16-239	16-268	16-272*	16-306	16-416*	16-424*	17-23	25-587	28-50	34-174				
RMDS	4-702#	10-86	10-98											
RMDSI	7-0#	13-F71	13-F73	13-F75	13-G26	13-G28	13-G30	15-122	15-180	16-115	16-149	24-104	25-34	25-36
	25-81	25-98	25-132	25-136	25-192	25-219	25-354	25-654	28-96	28-151	28-160	28-167	28-171	28-173
	28-182	28-186	28-188	28-197	28-199	28-201	28-211	28-215	28-217	30-124	31-23	31-25	31-27	31-37
	31-39	31-41	32-68	32-106	32-144	32-153	32-157	32-159	32-169	32-173	32-174	32-185	32-187	32-189
	32-201	32-203	32-205	32-216	32-220	32-222	33-28	34-88	34-90	34-119	34-256	34-322	34-457	34-459
	34-461	34-508	34-515	34-519	34-521	34-532	34-536	34-538	34-548	34-552	34-554	35-34	35-40	35-44
	35-46	35-56	35-60	35-61*	35-62	35-72	35-76	35-78	59-13					
RMDSO	7-0#													
RMDT	4-707#	10-92	10-94	43-26										
RMDTI	7-0#	59-5	59-17											
RMDTO	7-0#													
RMEC1	4-714#													
RMEC1I	7-0#	18-32	59-14											
RMEC10	7-0#													
RMEC2	4-715#	19-15												
RMEC2I	7-0#	18-53	19-14	30-92	33-94	59-15								
RMEC20	7-0#													
RMER1	4-703#	13-F48	13-G03	24-143										
RMER1I	7-0#	13-A11	13-A16	13-A40	13-A46	13-B19	13-B24	13-B48	13-B54	13-C92	13-C94	13-D58	13-D60	16-221
	18-23	18-25	24-135	24-150	24-152	25-76	25-152	25-179	25-222	25-237	25-280	25-357	25-383	25-398
	25-413	25-497	25-516	25-519	25-651	25-667	25-682	25-694	26-24	28-30	28-37	28-71	28-145	28-148
	28-169	30-69	31-51	31-55	31-57	31-58	31-69	31-71	31-72	31-83	31-85	31-86	31-97	31-99
	31-100	31-111	31-113	31-114	32-24	32-29	32-33	32-35	32-45	32-47	32-49	32-61	32-63	32-65
	32-78	32-80	32-82	32-155	32-232	32-236	32-238	32-240	32-250	32-252	32-254	32-264	32-268	32-270
	33-41	34-41	34-45	34-47	34-83	34-85	34-87	34-128	34-131	34-133	34-135	34-145	34-147	34-149
	34-159	34-161	34-163	34-193	34-283	34-286	34-290	34-292	34-301	34-303	34-305	34-315	34-317	34-318
	34-352	34-356	34-358	34-360	34-369	34-371	34-373	34-382	34-384	34-386	34-395	34-397	34-399	34-417
	34-419	34-421	34-427	34-483	34-485	34-487	34-534	35-42	59-13					
RMER10	7-0#	13-F48*	13-G03*	24-139										
RMER2	4-713#													
RMER2I	7-0#	13-=91	13->56	13-?29	13-?78	13-A27	13-A32	13-A33	13-A52	13-A57	13-B35	13-B40	13-B41	13-B60
	13-B65	13-C98	13-C99	13-D06	13-D64	13-D65	13-D72	13-E53	13-E58	13-E59	13-F13	13-F18	13-F20	15-55
	16-83	16-224	16-231	24-137	25-78	25-196	25-252	25-428	25-459	26-27	28-32	28-90	28-93	28-104
	28-132	28-135	28-184	28-213	30-114	32-31	32-92	32-99	32-101	32-103	32-116	32-118	32-120	32-130
	32-132	32-134	32-171	32-218	32-266	33-103	34-43	34-111	34-114	34-116	34-207	34-211	34-236	34-240
	34-242	34-244	34-254	34-258	34-260	34-270	34-272	34-274	34-288	34-470	34-472	34-474	34-517	34-550
	35-58	35-74	35-88	35-94	35-96	35-98	35-108	35-110	35-112	59-13				
RMER20	7-0#													
RMHR	4-711#													
RMHRI	7-0#													
RMHRO	7-0#													
RMLA	4-705#													
RMLAI	7-0#	59-16												
RMLAO	7-0#													
RMMR1	4-706#	13-E22	13-E30	13-E81	13-E90									
RMMR1I	7-0#	30-79	33-69	59-17										
RMMR10	7-0#	13-E22*	13-E28*	13-E81*	13-E88*									
RMMR2	4-712#													

RMMR2I	7-0#	30-101	33-81	59-17										
RMMR2O	7-0#													
RMOF	4-709#	13-64	13-104	13-148	13-211	13-251	13-294	13-352	13-392	13-436	13-527	13-570	13-614	13-675
	13-718	13-761	13-820	13-863	13-907	13-:00	13-:49	13-:92	13-;55	13-<15	13-<76	13-=43	13-=72	13->36
	13-?10	13-?58	13-a37	13-a77	13-A86	13-C28	13-C71	13-D36	13-E33	13-E92	13-F57	13-G11	13-G78	13-H18
	13-H62	13-I24	13-I66	13-J10	13-J73	13-K09	13-K65	13-L48	16-66	21-51				
RMOFI	7-0#	18-21	18-27	25-373	25-447	25-449	25-514	25-623	34-348	34-424	59-14	59-16		
RMOFO	7-0#	13-49*	13-196*	13-338*	13-512*	13-660*	13-805*	13-974*	13-;17	13-;19*	13-;40*	13-=17*	13->66	13->69*
	13->82*	13-?88	13-?92*	13-a14*	13-C05*	13-C64*	13-E08*	13-F39*	13-G63*	13-I09*	13-J58*	13-L09	13-L11*	13-L21*
	16-59*	16-235*	16-274*	16-283*	16-375	16-401	17-32	17-37	25-599	25-626	28-59	28-63	34-183	34-187
	59-11													
RMR	4-586#	31-51	31-83	31-87	32-232	32-250	32-253	34-128	34-159	34-162				
RMSN	4-708#													
RMSNI	7-0#	59-17												
RMSNO	7-0#													
RMWC	4-775#	13-10*	13-11	13-63	13-106	13-151	13-210	13-253	13-297	13-351	13-394	13-439	13-526	13-572
	13-617	13-674	13-720	13-764	13-819	13-865	13-910	13-999	13-:51	13-:95	13-;54	13-<17	13-<79	13-=42
	13-=74	13->39	13-?12	13-?61	13-a36	13-a79	13-A89	13-C27	13-C73	13-D39	13-E34	13-E95	13-F59	13-G14
	13-G77	13-H20	13-H65	13-I23	13-I68	13-J13	13-J72	13-K11	13-K68	13-L49	16-65			
RMWCI	7-0#	18-72	25-538	25-551	25-572	25-677	59-14							
RMWCO	7-0#	13-48*	13-99*	13-195*	13-246*	13-337*	13-386*	13-511*	13-565*	13-659*	13-713*	13-804*	13-857*	13-981*
	13-:25*	13-;39*	13-<09*	13-=21*	13->85*	13-a13*	13-a72*	13-A82*	13-C04*	13-E07*	13-F38*	13-G62*	13-H12*	13-I08*
	13-I61*	13-J57*	13-J96*	13-L22*	16-58*	16-276*	16-309	17-22	18-73	25-552	25-573			
RQA	4-670#	30-102	33-82											
RQB	4-671#	30-102	33-82											
RTC	4-515#													
SA1	4-551#													
SA16	4-547#													
SA2	4-550#													
SA4	4-549#													
SA8	4-548#													
SADMSK	4-555#	25-596												
SAVREG	43-13	46-1#												
SC	4-726#	24-90	24-107	24-123	30-30									
SC0	4-604#													
SC1	4-603#													
SC2	4-602#													
SC3	4-601#													
SC4	4-600#													
SCOPE	4-492#	13-2	13-33	13-180	13-322	13-496	13-644	13-789	13-968	13-;23	13-=09	13->73	13-?96	13-B87
	13-D97	13-F29	13-G47	13-H92	13-J43	13-L15	14-9							
SCTMSG	13-52	13-199	13-341	13-515	13-663	13-808	13-988	13-;43	13-a17	13-C08	13-E11	13-G66	13-I12	13-J61
	13-L26	49-3#												
SCTMSK	4-606#													
SEARCH	4-521#	25-100	25-533											
SECERR	13-126	13-171	13-273	13-317	13-414	13-488	13-592	13-637	13-740	13-784	13-885	13-960	13-:71	13-;15
	13-<35	13-<97	13-=99	13->64	13-?37	13-?86	13-A64	13-B72	13-D14	13-D80	13-E64	13-F24	13-F86	13-G42
	13-H40	13-H85	13-I88	13-J33	13-K46	13-L07	13-L67	25-24#						
SEEK	4-510#	13-;89	13-<51	13-=37	13->14									
SEKSTS	13-<03	13-<65	13-=60	13->28	28-19#									
SHUT	14-16	14-20	14-22#	44-6										
SHUT2	44-1	44-1	44-3#											
SIZCLK	12-28	22-3#												
SKI	4-683#	15-178	16-147	28-49	28-52	28-75	28-105	28-107	28-111	28-122	28-184	30-115	32-92	32-116
	32-119	32-218	34-173	34-176	34-197	34-208	34-210	34-211	34-213	34-225	34-517	35-74	35-89	35-108
	35-111													

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

```
        47-1*
SWREG    5-1#    10-23   10-28   44-1    44-1    44-1
SYSTAT  10-76    49-25#
TA1      4-544#  12-57
TA2      4-543#  13-46   13-193  13-335  13-509  13-657  13-802  13-979  13-;37  13-=16  13->68  13-@11  13-C02  13-E05
        13-F36   13-G60  13-J55  13-L19
TA4      4-542#  12-57
TA8      4-541#  12-57
TADMSK   4-554#  25-594
TAG      4-672#
TAGADR   5-9#     6-0
TAP      4-648#
TBITVE   4-492#
TIMOUT  13-73    13-115  13-160  13-220  13-262  13-306  13-361  13-403  13-448  13-538  13-581  13-626  13-686  13-729
        13-773   13-831  13-874  13-919  13-:11  13-:60  13-;04  13-;66  13-;96  13-<24  13-<58  13-<86  13-=53  13-=83
        13->21   13->48  13-?21  13-?70  13-@46  13-@88  13-A98  13-C39  13-C82  13-D48  13-E44  13-F04  13-F79  13-G35
        13-G87   13-H29  13-H74  13-I35  13-I77  13-J22  13-J82  13-K23  13-K35  13-K83  13-K96  13-L59  15-136  15-194
        16-95    16-127  16-161  16-190  23-9#
TKVEC    4-492#  44-1*   44-1*
TPVEC    4-492#
TRAPVE   4-492#  10-23*  10-23*
TRE      4-727#  16-208  24-90   24-107  24-123  25-102  25-104  25-540
TRTVEC   4-492#
TST      4-673#  30-103  33-83
TST1    13-2#    41-1
TST10   13-968#  41-1
TST11   13-:23#  41-1
TST12   13-=09#  41-1
TST13   13->73#  41-1
TST14   13-?96#  41-1
TST15   13-@87#  41-1
TST16   13-D97#  41-1
TST17   13-F29#  41-1
TST2    13-33#   41-1
TST20   13-G47#  41-1
TST21   13-H92#  41-1
TST22   13-J43#  41-1
TST23   13-L15#  41-1
TST3    13-180#  41-1
TST4    13-322#  41-1
TST5    13-496#  41-1
TST6    13-644#  41-1
TST7    13-789#  41-1
TSTNMB  43-36*   43-37*  43-39   43-151#
TSTPRP  13-41    13-93   13-139  13-188  13-240  13-286  13-330  13-381  13-428  13-504  13-559  13-605  13-652  13-707
        13-753   13-797  13-852  13-899  13-985  13-:40  13-:84  13-;31  13-;86  13-<48  13-=33  13->12  13->98  13-?50
        13-@06   13-@66  13-A76  13-B97  13-C59  13-D27  13-E18  13-E77  13-F44  13-F99  13-G55  13-H07  13-H53  13-I00
        13-I55   13-J01  13-J50  13-K02  13-K58  13-L40  15-38#
TSTQUE   7-0#    12-8    12-9*   12-51   13-2    13-33   13-180  13-322  13-496  13-644  13-789  13-968  13-:23  13-=09
        13->73   13-?96  13-@87  13-D97  13-F29  13-G47  13-H92  13-J43  13-L15  14-11   14-13*  14-17   14-17*  27-24
        29-17    30-54   33-53
TYPBN   43-141   46-1#
TYPDS   14-20    14-20   43-135  46-1#
TYPE    10-6     10-28   10-42   10-48   10-53   10-59   10-76   10-80   10-82   10-96   10-102  10-105  10-109  10-117
        10-120   10-121  10-122  10-129  11-11   11-16   11-17   11-19   11-20   11-28   11-32   11-34   11-40   11-44
        11-48    11-54   11-61   11-62   11-64   11-69   11-75   11-86   11-87   11-92   11-100  12-3    12-6    12-7
```

CROSS REFERENCE TABLE (CREF V04.00)

	12-14	12-25	12-30	12-60	12-61	13-52	13-199	13-341	13-515	13-663	13-808	13-988	13-;43	13-a17
	13-C08	13-E11	13-G66	13-I12	13-J61	13-L26	14-20	14-20	14-20	37-1	38-1	39-1	40-1	42-1
	42-1	43-20	43-21	43-31	43-32	43-38	43-43	43-45	43-53	43-96	43-102	43-106	43-118	43-124
	43-126	43-144	43-146	44-1	44-1	44-1	44-1	44-1	44-1	44-1	44-1	44-1	44-1	44-1
	44-1	44-1	44-1	44-1	44-1	44-1	44-1	46-1#	47-1					
TYPOC	10-8	11-33	11-47	43-138	44-1	46-1#								
TYPON	46-1#													
TYPOS	10-45	10-56	10-81	12-16	12-62	43-22	43-39	43-44	43-46	46-1#				
U0	4-752#													
U1	4-752#													
U2	4-752#													
UNS	4-574#	13-F48	13-G03	30-70	31-51	31-55	31-59	32-232	32-264	32-269	34-283	34-286	34-291	
UNTMSK	4-756#	24-48	24-50											
UNTOFF	10-109	49-31#												
UNTON	10-120	49-32#												
UPE	4-739#	25-266												
USE	4-695#	13-A06	13-B14	17-29										
USRFIL	16-279	16-282	16-333	61-12#										
VV	4-568#	15-122	16-115	25-192	28-96	28-161	28-162	28-211	28-216	31-23	31-26	32-106	32-145	32-146
	32-169	32-175	33-29	34-90	34-119	34-509	34-510	34-548	34-553	35-35	35-36	35-56	35-61	
WC	4-638#	30-80	33-70											
WCD	4-525#	13-291	13-433	13-758	13-904	13-:89	13->33	13-?55	13-K86					
WCE	4-738#	13-457	13-463	13-928	13-933	25-165	50-77	50-78						
WCEHI	4-763#													
WCELO	4-764#													
WCF	4-583#	4-590	25-236	34-483	34-486									
WCH	4-526#													
WD	4-529#	13-100	13-247	13-388	13-566	13-714	13-859	13-:27	13-:45	13-<11	13-a73	13-C66	13-E09	13-E27
	13-F40	13-F53	13-H14	13-I62										
WH	4-530#	13-50	13-197	13-339	13-513	13-661	13-806	13-982	13-;41	13-a15	13-C06	13-G64	13-I10	13-J59
	13-L24	13-L44	34-346											
WLE	4-577#	4-590	25-217	25-221	25-234	25-249	25-264	34-283	34-315	34-319	34-327	50-81	50-82	
WRL	4-563#	25-219	34-322											
XSIZ	10-70#	11-71	11-101											
XXDP	7-0#	10-33*	10-36*	10-37	10-39*	10-44	10-55	10-113	10-115					
Y	49-36#													
ZEROS	13-54	13-201	13-343	13-990	13-a19	13-J63	52-38#							

CROSS REFERENCE TABLE (CREF V04.00)

```
$$CMRE    5-10#
$$CMTM    5-10#    6-0      6-0      6-0      6-0      6-0
$$ESCA    4-492#
$$NEWT    4-492#   13-2     13-33    13-180   13-322   13-496   13-644   13-789   13-968   13-;23   13-=09   13->73   13-?96   13-B87
         13-D97    13-F29   13-G47   13-H92   13-J43   13-L15
$$SET    46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1     46-1#
$$SETM   10-23    10-23#
$$SKIP    4-492#
.$ACT1    4-484#   5-5
.$APTB    4-484#   6-0      6-0#
.$APTH    4-484#   5-8
.$APTY    4-484#  48-1
.$CATC    4-480#   5-1
.$CMTA    4-481#   5-10
.$EOP     4-481#  14-20
.$ERRO    4-481#  42-1
.$POWE    4-483#  47-1
.$RDDE    4-482#
.$RDOC    4-482#  45-1
.$READ    4-482#  44-1
.$SAVE    4-483#  36-1
.$SCOP    4-481#  41-1
.$SIZE    4-483#
.$TRAP    4-483#  46-1
.$TYPB    4-482#  37-1
.$TYPD    4-482#  38-1
.$TYPE    4-481#  40-1
.$TYPO    4-482#  39-1
.EQUAT    4-480#   4-492
.HEADE    4-480#   4-488
.SETUP    4-480#   4-785
.SWRHI    4-480#   4-489
.SWRLO    4-480#   4-489#   4-490
CALCLR    4-192#
CALSUB    4-206#  13-41    13-52    13-68    13-71    13-73    13-76    13-80    13-93    13-110   13-113   13-115   13-118   13-122
         13-124   13-126   13-139   13-155   13-158   13-160   13-163   13-167   13-169   13-171   13-175   13-188   13-199   13-215
         13-218   13-220   13-223   13-227   13-240   13-257   13-260   13-262   13-265   13-269   13-271   13-273   13-286   13-301
         13-304   13-306   13-309   13-313   13-315   13-317   13-330   13-341   13-356   13-359   13-361   13-364   13-368   13-381
         13-398   13-401   13-403   13-406   13-410   13-412   13-414   13-428   13-443   13-446   13-448   13-451   13-455   13-460
         13-488   13-504   13-515   13-533   13-536   13-538   13-541   13-545   13-559   13-576   13-579   13-581   13-584   13-588
         13-590   13-592   13-605   13-621   13-624   13-626   13-629   13-633   13-635   13-637   13-639   13-652   13-663   13-681
         13-684   13-686   13-689   13-693   13-707   13-724   13-727   13-729   13-732   13-736   13-738   13-740   13-753   13-768
         13-771   13-773   13-776   13-780   13-782   13-784   13-797   13-808   13-826   13-829   13-831   13-834   13-838   13-852
         13-869   13-872   13-874   13-877   13-881   13-883   13-885   13-899   13-914   13-917   13-91?   13-922   13-926   13-930
         13-960   13-985   13-988   13-:06   13-:09   13-:11   13-:14   13-:18   13-:40   13-:55   13-:58   13-:60   13-:63   13-:67
         13-:69   13-:71   13-:84   13-:99   13-;02   13-;04   13-;07   13-;11   13-;13   13-;15   13-;31   13-;43   13-;61   13-;64
         13-;66   13-;69   13-;73   13-;86   13-;91   13-;94   13-;96   13-;99   13-<03   13-<21   13-<24   13-<27   13-<31   13-<33
         13-<35   13-<48   13-<53   13-<56   13-<58   13-<61   13-<65   13-<83   13-<86   13-<89   13-<93   13-<95   13-<97   13-<99
         13-=33   13-=48   13-=51   13-=53   13-=56   13-=60   13-=78   13-=81   13-=83   13-=86   13-=95   13-=97   13-=99   13->12
         13->16   13->19   13->21   13->24   13->28   13->43   13->46   13->48   13->51   13->60   13->62   13->64   13->98   13-?16
         13-?19   13-?21   13-?24   13-?33   13-?35   13-?37   13-?50   13-?65   13-?68   13-?70   13-?73   13-?82   13-?84   13-?86
         13-@06   13-@17   13-@41   13-@44   13-@46   13-@49   13-@53   13-@66   13-@83   13-@86   13-@88   13-@91   13-@95   13-A14
         13-A30   13-A43   13-A55   13-A64   13-A76   13-A93   13-A96   13-A98   13-B01   13-B05   13-B22   13-B38   13-B51   13-B63
         13-B72   13-B97   13-C08   13-C34   13-C37   13-C39   13-C42   13-C46   13-C59   13-C77   13-C80   13-C82   13-C85   13-C89
         13-D12   13-D14   13-D27   13-D43   13-D46   13-D48   13-D51   13-D55   13-D78   13-D80   13-D82   13-E11   13-E18   13-E39
         13-E42   13-E44   13-E47   13-E51   13-E56   13-E64   13-E77   13-E99   13-F02   13-F04   13-F07   13-F11   13-F16   13-F24
```

CROSS REFERENCE TABLE (CREF V04.00)