RM05/3/2 RM05/3/2 FCTNL TST1 AH-F922A-MC
CZRMMA0 FICHE 1 OF 2

RM05/3/2 RM05/3/2 FCTNL TST1 AH-F922A-MC
CZRMMA0 FICHE 2 OF 2

```
.REM    \
```

IDENTIFICATION

PRODUCT CODE: AC-F921A-MC

PRODUCT NAME: CZRMMA0 RM05/3/2 FCTNL TST 1

DATE CREATED: APRIL 1980

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

CONTENTS

1.0 INTRODUCTION

1.1 ABSTRACT

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

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

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

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

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

1.2 UNIT UNDER TEST

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

2.0 OPERATING REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

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

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

2.2 MEDIA REQUIREMENTS

EACH UNIT BEING TESTED MUST BE LOADED WITH A SCRATCH DISK PACK BEFORE TESTING BEGINS ON THAT UNIT. THE DISK MAY BE FORMATTED OR UNFORMATTED.

2.3 PREREQUISITE DIAGNOSTIC PROGRAMS

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

3.0 OPERATING PROCEDURE

3.1 LOADING

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

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

3.2 SWITCH OPTIONS

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

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

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

3.3 STARTING

THE PROGRAM STARTS AT LOCATION 200, WHICH PROVIDES WORST CASE TEST CONDITIONS IF RUNNING IN AN AUTOMATIC ENVIRONMENT. IF RUNNING IN A STAND-ALONE ENVIRONMENT, THE PROGRAM USES CONSOLE DIALOGUE TO ALLOW THE OPERATOR TO CONTROL TEST CONDITIONS.

3.4 HALTING

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

3.5 RESTARTING

THE PROGRAM CAN BE RESTARTED AT ADDRESS 200.

4.0 OPERATOR INTERFACE

4.1 PROGRAM ID

THE PROGRAM TYPES ITS TITLE AND MAINDEC NUMBER THE FIRST TIME IT IS STARTED AFTER BEING LOADED. PROGRAM IDENTIFICATION DOES 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..

THE FIRST QUESTION TYPED OUT IS: "TYPE HELP TEXT (Y/N) ?". IF THE OPERATOR RESPONDS WITH A "Y", THE PROGRAM WILL TYPE A BRIEF HELP MESSAGE WHICH WILL LIST SWITCH OPTIONS, ETC.

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

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

THE THIRD QUESTION TYPED IS, 'TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S) AND TERMINATE INPUT WITH A CARRIAGE RETURN.' THEN, 'DRIVE(S):' IS TYPED AND WAITS FOR THE OPERATOR TO TYPE AN "A", TO TEST ALL POSSIBLE DRIVES OR THE NUMBER(S) OF THE DRIVE(S) HE WANTS TESTED AND TERMINATE HIS INPUT WITH A "CARRIAGE RETURN".

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

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

4.3 PROGRESS REPORTS

AN END OF PASS REPORT OCCURS EACH TIME THE PROGRAM IS EXECUTED FOR ALL DEVICES IN THE TEST QUE. THE END OF PASS RFPORT INCLUDES A MESSAGE AND AN ERROR SUMMARY.

4.4 PERFORMANCE REPORT

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

4.5 PROGRAM HALTS

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

4.6 ERROR REPORTS

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

5.0 ENVIRONMENTAL SUPPORT

5.1 PROCESSOR COMPATIBILITY

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

5.2 DUAL PORT CONFIGURATIONS

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

5.3 MEMORY PARITY HARDWARE

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

5.4 MEMORY MANAGEMENT HARDWARE

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

5.5 ACT11, APT11 COMPATIBILITY

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

5.6 XXDP COMPATIBILITY

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

5.7 OPERATING SYSTEM COMPATIBILITY

THE PROGRAM IS NOT COMPATIBLE WITH ANY SOFTWARE OPERATING SYSTEM.

6.0 TEST DESCRIPTION

CONTROLLER ACCESS TEST

PURPOSE:

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

PROCEDURE:

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

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

THE END OF PASS HANDLER.

DEVICE AVAILABLE TEST

PURPOSE:

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

PROCEDURE:

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

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

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

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

.DEVICE AVAILABLE - THE DEVICE IS AVAILABLE FOR TESTING.

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

DRIVE TYPE TEST

PURPOSE:

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

PROCEDURE:

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

```
343
344                          UNIBUS INITIALIZE TEST
345
346                               PURPOSE:
347
348                                    TO VERIFY THAT ALL APPLICABLE SUBSYSTEM REGISTERS ARE
349                               INITIALIZED BY THE RESET INSTRUCTION.
350
351                               PROCEDURE:
352
353                                    NONZERO VALUES ARE WRITTEN IN EACH APPLICABLE REGISTER.  A
354                               RESET INSTRUCTION IS EXECUTED, AND THE REGISTERS ARE TESTED TO
355                               INSURE THEY WERE INITIALIZED.  THIS TEST IS DONE ONCE BECAUSE OF
356                               APT COMPATIBILITY REQUIREMENTS.
357
358                                    THE FOLLOWING REGISTERS ARE PRESET BEFORE THE INITIALIZE
359                               OCCURS:
360
361                                         RMCS1 - 003577
362
363                                         RMBA  - 777776
364
365                                         RMCS2 - 021037
366
367                                         RMER1 - 777777
368
369                                         RMER2 - 777777
370
371                                         RMMR  - 040001
372
373                                    IN ADDITION, THE DATA BUFFER IS USED TO FORCE DLT,TRE,SC AND OR
374                               TO A ONE AND TO FORCE IR TO A ZERO.
375
376
377
378
379
380                          CONTROLLER CLEAR TEST
381
382                               PURPOSE:
383
384                                    TO VERIFY THAT APPLICABLE SUBSYSTEM REGISTERS ARE
385                               INITIALIZED BY A "CONTROLLER CLEAR" OPERATION.
386
387                               PROCEDURE:
388
389                                    LIKE THE UNIBUS INITIALIZE TEST, THIS TEST WRITES NONZERO
390                               VALUES IN THOSE REGISTERS WHICH ARE INITIALIZED BY CONTROLLER
391                               CLEAR.  THE SUBSYSTEM IS THEN CLEARED USING A CONTROLLER CLEAR,
392                               I.E., BIT 5 OF CONTROL STATUS REGISTER 2 (RMCS2), AND EACH
393                               REGISTER IS READ TO INSURE IT WAS CLEARED.
394
395
396
397
398
399                          ERROR CLEAR TEST
```

PURPOSE:

TO VERIFY THAT ALL APPLICABLE RH MASSBUS CONTROLLER STATUS AND ERROR CONDITIONS ARE INITIALIZED BY AN ERROR CLEAR OPERATION.

PROCEDURE:

AN "RH ERROR CLEAR" OPERATION, I.E., WRITING A ONE IN TRE, BIT 14 OF RMCS1 WILL CLEAR THE FOLLOWING STATUS BITS:

.TRE, BIT 14 OF RMCS1

.MCPE, BIT 13 OF RMCS1 - READ ONLY

.DLT, BIT 15 OF RMCS2 - READ ONLY

.WCE, BIT 14 OF RMCS2 - READ ONLY

.UPE, BIT 13 OF RMCS2

.NED, BIT 12 OF RMCS2 - READ ONLY

.PGE, BIT 10 OF RMCS2 - READ ONLY

.MXF, BIT 09 OF RMCS2

.MDPE, BIT 08 OF RMCS2 - READ ONLY

THE TEST SETS UPE AND MXF STATUS BITS, THEN SETS TRE (ERROR CLEAR) AND VERIFIES THAT ALL THE ABOVE STATUS BITS ARE CLEARED.

DRIVE STATUS TEST

PURPOSE:

TO VERIFY THAT THE STORAGE MODULE DISK DRIVE IS IN A STATE THAT PERMITS FURTHER TESTING.

PROCEDURE:

THIS TEST INITIALIZES THE MASSBUS AND EXAMINES STATUS OF THE SELECTED DEVICE FOR THE FOLLOWING CONDITIONS:

.MOL, BIT 12 OF RMDS =1, INDICATING THAT UNIT READY IS ASSERTED BY THE DRIVE;

.WRL, BIT 11 OF RMDS =0, INDICATING THAT THE DRIVE IS NOT IN A WRITE PROTECT STATE;

.DVC, BIT 07 OF RMER3 =0, INDICATING DRIVE FAULT IS UNASSERTED BY THE DRIVE;

.UNS, BIT 14 OF RMER1 SHOULD EQUAL DVC, OTHERWISE AC POWER IS LOW OR A FAILURE HAS OCCURRED WITH UNSAFE STATUS.

PRIMARY/SECONDARY ERROR TEST

PURPOSE:

TO VERIFY THAT THE RM05/3/2 CAN EXECUTE A COMMAND WITHOUT INCURRING UNEXPECTED ERRORS.

PROCEDURE:

THE TEST EXECUTES A PACK ACKNOWLEDGE COMMAND AND MAKES SURE THAT GO RESETS AND THAT THERE ARE NO PARITY ERPORS, ETC. VOLUME VALID STATUS IS IGNORED.

DIAGNOSTIC MODE TEST

PURPOSE:

TO VERIFY THAT MAINTENANCE HARDWARE IS OPERATIONAL.

PROCEDURE:

THE TEST THAT DIAGNOSTIC MODE CAN BE SET AND RESET, THEN VERIFIES THAT "MOL, PIP, WRL, SKI, AND DVC" CAN BE CONTROLLED USING MAINTENANCE REGISTER 1.

PACK ACKNOWLEDGE TEST

PURPOSE:

TO VERIFY THAT VOLUME VALID CAN BE SET BY A PACK ACKNOWLEDGE COMMAND, LENDING CREDENCE TO THE EXECUTION OF THE COMMAND AND TO THE STABILITY OF THE UNIT READY SIGNAL FROM THE DRIVE.

PROCEDURE:

A PACK ACKNOWLEDGE COMMAND IS ISSUED TO THE SELECTED DEVICE AND VOLUME VALID STATUS, BIT 10 OF RMDS, IS CHECKED FOR ONE.

```
514
515                         RECALIBRATE TEST
516
517                              PURPOSE:
518
519                                   THE PRIMARY PURPOSE IS TO ASCERTAIN THAT THE DRIVE WILL
520                              EXECUTE A RECALIBRATE OPERATION TO THE EXTENT THAT "PIP" AND
521                              "SKI" STATUS BECOME UNASSERTED AT THE COMPLETION OF THE
522                              RECALIBRATE.  THE SECONDARY PURPOSE IS TO PUT THE DRIVE IN A
523                              KNOWN STATE SO THAT FURTHER TESTS CAN CHECK FOR UNEXPECTED STATE
524                              CHANGES IN THE DRIVE.
525
526                              PROCEDURE:
527
528                                   THE RECALIBRATE TEST PRESETS THE DISK ADDRESS REGISTER,
529                              RMDA, AND THE DESIRED CYLINDER REGISTER, RMDC, TO ZERO, THEN
530                              EXECUTES A RECALIBRATE COMMAND.  THE TEST VERIFIES THE FOLLOWING
531                              CONDITIONS:
532
533                                   .SKI=0, "SEEK ERROR" IS INACTIVE;
534
535                                   .PIP=0, "ON CYLINDER" IS ACTIVE, AS INDICATED BY
536                              "POSITIONING IN PROGRESS" BEING INACTIVE;
537
538                                   .IAE=0, NO "INVALID ADDRESS ERROR" DURING RECALIBRATE;
539
540                                   .OPI=0, NO "OPERATION INCOMPLETE ERROR", INDICATING THAT
541                              THAT THE DRIVE WAS READY AI THE START OF THE COMMAND AND THAT ON
542                              CYLINDER STATUS WENT INACTIVE WHEN THE DRIVE RECEIVED THE
543                              COMMAND;
544
545                                   .ATA=1, ATTENTION IS SET BY RECALIBRATE.
546
547
548
549
550
551                         ABORT RECALIBRATE TEST
552
553                              PURPOSE:
554
555                                   TO VERIFY THAT THE RM05/3/2 INHIBITS A RECALIBRATE WHEN AN
556                              ABORT CONDITION EXISTS AT THE START OF THE COMMAND.
557
558                              PROCEDURE:
559
560                                   THE TEST SETS AN ERROR IN THE ERROR REGISTER AND ISSUES A
561                              RECALIBRATE COMMAND, VERIFYING THAT "PIP" REMAINS INACTIVE.
562
563
564
565
566
567                         IVC RECALIBRATE TEST
568
569                              PURPOSE:
570
```

TO VERIFY THAT INVALID COMMAND STATUS, BIT 12 OF RMER2, SETS WHEN VOLUME VALID IS INACTIVE DURING A RECALIBRATE COMMAND.

PROCEDURE:

THE PROGRAM SETS AND RESETS DIAGNOSTIC MODE WHICH CAUSES VOLUME VALID, BIT 6 OF RMDS TO RESET. THE PROGRAM THEN EXECUTES A RECALIBRATE COMMAND AND VERIFIES THAT "IVC" STATUS SETS AND THAT "PIP" REMAINS INACTIVE.

IAE RECALIBRATE TEST

PURPOSE:

TO VERIFY THAT INVALID ADDRESS ERROR DOES NOT SET DURING A RECALIBRATE COMMAND .

PROCEDURE:

THE TEST PRESETS THE DISK ADDRESS (RMDA) AND THE DESIRED CYLINDER ADDRESS (RMDC) TO ILLEGAL VALUES AND ISSUES A RECALIBRATE COMMAND, VERIFYING THAT "IAE" DOES NOT SET DURING THE COMMAND.

RECALIBRATE AT OFFSET

PURPOSE:

TO VERIFY THAT OFFSET MODE DOES NOT AFFECT RECALIBRATE.

PROCEDURE:

THE TEST EXECUTES AN OFFSET COMMAND, THEN EXECUTES A RECALIBRATE COMMAND AND VERIFIES THAT THERE ARE NO ERRORS DURING RECALIBRATE.

DRIVE CLEAR TEST

PURPOSE:

TO VERIFY THAT ALL APPLICABLE RM05/3/2 MASSBUS ADAPTER ERROR AND STATUS CONDITIONS ARE INITIALIZED BY A "DRIVE CLEAR" COMMAND.

PROCEDURE:

THIS TEST WRITES ONES IN THOSE MASSBUS ADAPTER BITS WHICH ARE INITIALIZED BY DRIVE CLEAR , THEN ISSUES THE DRIVE CLEAR COMMAND AND VERIFIES THAT EACH BIT IS CLEARED. ADDITIONALLY, REGISTERS WHICH ARE NOT AFFECTED BY "DRIVE CLEAR" ARE ALSO PRESET AND VERIFIED.

THE FOLLOWING ITEMS ARE PRESET:

.RMER1, ERROR REGISTER 1 IS SET TO ALL ONES;

.DMD, DIAGNOSTIC MODE IS SET;

.RMER2, ERROR REGISTER 3, IS SET TO ALL ONES.

FOLLOWING THE DRIVE CLEAR COMMAND, THE FOLLOWING ITEMS ARE CHECKED:

.RMCS1, IS CHECKED FOR DVA=1, F0-F4 AND GO=0, ALL OTHER BITS ARE DONT CARES;

.RMDS, IS CHECKED FOR 0, EXCEPT FOR MOL,DPR,DRY, AND VV WHICH SHOULD BE ONE, AND PGM WHICH IS A DONT CARE.

.RMER1, IS CHECKED FOR 0;

.RMAS, IS CHECKED TO INSURE THE APPROPRIATE ATA BIT IS 0;

.RMMR, IS CHECKED FOR 0, EXCEPT FOR WORD CLOCK, LAST SECTOR, AND LAST SECTOR AND TRACK WHICH ARE DONT CARES;

.RMMR2 IS CHECKED FOR 0, EXCEPT FOR THE TEST BIT WHICH IS ONE, AND REQA, REQB WHICH ARE DONT CARES;

.RMEC2 IS CHECKED FOR 0;

.RMER3 IS CHECKED FOR 0;

NOP TEST

PURPOSE:

TO VERIFY THE EXECUTION OF "NOP" COMMAND.

PROCEDURE:

A NOP COMMAND IS EXECUTED ON THE SELECTED DEVICE AND STATUS IS CHECKED TO VERIFY THAT THERE WERE NO ERRORS OR UNEXPECTED CHANGES IN STATUS.

OFFSET TEST

PURPOSE:

T VERIFY THE EXECUTION OF ''OFFSET'' COMMAND.

PROCEDURE:

THE OFFSET COMMAND IS EXECUTED AND THE PROGRAM CHECKS THAT OFFSET STATUS, BIT 0 OF RMDS IS SET AND THAT ATTENTION, BIT 15 OF RMDS IS ALSO SET. ADDITIONALLY, CONTROLLER, ADAPTER, AND DRIVE STATUS IS CHECKED FOR UNEXPECTED CHANGES.

GO/ATA TEST

PURPOSE:

TO VERIFY THAT ''ATA'' WILL RESET WITH ''GO'' PROVIDING COMPOSITE ERROR IS INACTIVE.

PROCEDURE:

ATTENTION, BIT 15 OF RMDS, IS SET USING AN OFFSET COMMAND AND RESET USING A NOP COMMAND.

WRITE ATA TEST

PURPOSE:

TO VERIFY THAT ATTENTION CAN BE RESET BY WRITING THE ATTENTION SUMMARY REGISTER, RMAS.

PROCEDURE:

ATTENTION IS SET USING AN OFFSET COMMAND AFTER WHICH THE PROGRAM WRITES A 0 IN THE ATTENTION SUMMARY REGISTER, VERIFYING THT ATTENTION REMAINS SET. FOLLOWING THAT, THE PROGRAM WRITES A 1 IN RMAS AND VERIFIES THAT ATTENTION RESETS.

ERROR/ATA TEST

PURPOSE:

TO VERIFY THAT ''GO'' DOES NOT RESET ''ATA'' WHEN THERE IS A COMPOSITE ERROR.

PROCEDURE:

"ATA" IS SET WITH AN OFFSET COMMAND AFTER WHICH ONE OF THE ERROR BITS IS SET. THE PROGRAM THEN ISSUES A NOP COMMAND AND VERIFIES THAT THE ATTENTION BIT REMAINS SET.

PROGRAM INTERRUPT TEST

PURPOSE:

TO VERIFY THAT THE SUBSYSTEM WILL GENERATE A PROGRAM INTERRUPT WHEN PROCESSOR PRIORITY IS LESS THAN CONTROLLER PRIORITY AND INTERRUPT IS ENABLED.

PROCEDURE:

ATTENTION IS SET USING AN OFFSET COMMAND. WITH INTERRUPT ENABLED AND PROCESSOR PRIORITY SET BELOW CONTROLLER PRIORITY, THE PROGRAM VERIFIES THAT A PROGRAM INTERRUPT OCCURS.

INHIBIT INTERRUPT TEST

PURPOSE:

TO VERIFY THAT A PROGRAM INTERRUPT DOES NOT OCCUR WHEN (1) PROCESSOR AND CONTROLLER PRIORITY ARE THE SAME AND INTERRUPT IS ENABLED OR (2) WHEN PROCESSOR PRIORITY IS LESS THAN CONTROLLER AND INTERRUPTS ARE NOT ENABLED.

PROCEDURE:

ATTENTION IS SET USING AN OFFSET COMMAND WITH THE PRIORITY OF THE PROCESSOR SET EQUAL TO THE PRIORITY OF THE CONTROLLER. INTERRUPT IS ENABLED AND THE TEST VERIFIES THAT A PROGRAM IN'ERRUPT DOES NOT OCCUR. INTERRUPTS ARE DISABLED AND PROCESSOR PRIORITY IS LOWERED AND THE TEST VERIFIES THAT A PROGRAM INTERRUPT DOES NOT OCCUR.

RETURN TO CENTERLINE TEST

PURPOSE:

TO VERIFY THE EXECUTION OF "RETURN TO CENTERLINE" COMMAND.

```
                        PROCEDURE:

                             THIS TEST ISSUES AN RTC COMMAND  AND  VERIFIES  THAT  OFFSET
                        STATUS, BIT 0 OF RMDS IS RESET AND THAT ATTENTION, BIT 15 OF RMDS
                        IS SET.  UNEXPECTED STATUS OR ERROR CONDITIONS ARE ALSO VERIFIED.




                   READ IN PRESET TEST

                        PURPOSE:

                             TO VERIFY THE EXECUTION OF "READ IN PRESET" COMMAND.

                        PROCEDURE:

                             THIS TEST LOADS NON ZERO VALUES IN THOSE  ADAPTER  REGISTERS
                        WHICH  ARE  INITIALIZED  BY  THE  READ  IN  PRESET  COMMAND, THEN
                        EXECUTES THE COMMAND AND VERIFIES THE CONTENTS OF EACH  REGISTER.
                        THE FOLLOWING REGISTERS ARE CHECKED:

                             .RMDA,        THE DISK ADDRESS REGISTER, IS LOADED WITH  ALL
                        ONES AND VERIFIED TO BE ZERO AFTER THE RIP COMMAND;

                             .RMDC,        THE DESIRED CYLINDER REGISTER, IS LOADED  WITH
                        ALL ONES (001777) AND VERIFIED TO BE ZERO AFTER THE RIP COMMAND;

                             .RMOF,        THE OFFSET REGISTER, IS PRESET (TO 016200) AND
                        VERIFIED TO BE ZERO AFTER THE TEST;




                   RMDC CLEAR OFFSET TEST

                        PURPOSE:

                             TO VERIFY THAT WRITING THE DESIRED CYLINDER REGISTER  (RMDC)
                        WILL CLEAR OFFSET MODE.

                        PROCEDURE:

                             OFFSET MODE IS SET USING THE OFFSET COMMAND, THEN  RESET  BY
                        WRITING RMDC.




                   ILLEGAL FUNCTION TEST

                        PURPOSE:

                             TO VERIFY THAT THE RM05/3/2 SUBSYSTEM  DETECTS  ALL  ILLEGAL
```

FUNCTIONS.

PROCEDURE:

EACH ILLEGAL FUNCTION CODE IS EXECUTED WITH THE PROGRAM VERIFYING THAT "ILLEGAL FUNCTION" STATUS, BIT 0 OF RMER1 IS SET FOR EACH CODE. THE SUBSYSTEM IS INITIALIZED PRIOR TO EACH ILLEGAL FUNCTION TEST.

INVALID COMMAND TEST

PURPOSE:

TO VERIFY IVC ERROR DETECTION.

PROCEDURE:

THE TEST RESETS VOLUME VALID USING MAINTENANCE UNIT READY, THEN EXECUTES A NOP COMMAND AND VERIFIES THAT IVC IS SET. THE PROCESS IS REPEATED FOR EACH FUNCTION CODE, WITH IVC BEING CHECKED ACCORDING TO THE FUNCTION.

INVALID ADDRESS ERROR TEST

PURPOSE:

TO VERIFY IAE ERROR DETECTION.

PROCEDURE:

THE TEST EXECUTES EACH FUNCTION CODE WITH RMDA, AND RMDC SET TO ILLEGAL ADDRESSES, AND VERIFIES IAE ACCORDING TO THE FUNCTION.

WRITE LOCK ERROR TEST

PURPOSE:

TO VERIFY WLE ERROR DETECTION.

PROCEDURE:

THE TEST SIMULATES WRITE PROTECT USING MAINTENANCE WRITE PROTECT AND VERIFIES WLE ACCORDING TO THE FUNCTION BEING EXECUTED. EACH FUNCTION CODE IS TESTED.

ERROR ABORT TESTS

PURPOSE:

TO TEST COMMAND EXECUTION DURING AN ABORT CONDITION.

PROCEDURE:

EACH FUNCTION CODE IS EXECUTED UNDER A SIMULATED UNSAFE CONDITION AND THE TEST VERIFIES THAT GO IS RESET.

RMR TEST

PURPOSE:

TO VERIFY THAT RMR ERROR IS DETECTED.

PROCEDURE:

THE TEST EXECUTES A NOP COMMAND WITH DEBUG CLOCK ENABLED. WITH GO SET, THE TEST WRITES RMCSI, VERIFYING THAT RMR ERROR SETS.

PARITY ERROR TEST

PURPOSE:

TO VERIFY THAT PARITY ERRORS ARE DETECTED.

PROCEDURE:

WITH PAT SET TO CAUSE BAD PARITY ON THE CONTROL BUS, THE TEST WRITES A SHIFTING ONE BIT PATTERN TO RMDA AND VERIFIES THAT AN ERROR IS DETECTED BY THE RM05/3/2 FOR EACH PATTERN.

ILLEGAL REGISTER TEST

PURPOSE:

TO VERIFY THE DETECTION OF ILLEGAL REGISTER ADDRESSES BY THE SUSBSYSTEM.

PROCEDURE:

EACH REGISTER ADDRESS IS ACCESSED AND ILLEGAL REGISTER STATUS, BIT 1 OF RMER1 IS CHECKED ACCORDING TO THE ADDRESS USED. NOTE THAT THE EXECUTION OF THIS TEST IS DEPENDENT ON THE REGISTER ADDRESS JUMPER IN THE RH CONTROLLER BECAUSE THE RANGE OF ADDRESSES WHICH THE CONTROLLER WILL RESPOND TO IS LIMITED BY THE WAY THE JUMPER IS CUT.

SEEK TESTS

PURPOSE:

THE PURPOSE OF EACH OF THE FOLLOWING SEEK TESTS IS TO VERIFY THE EXECUTION OF SEEK OPERATIONS BY THE RM05/3/2 SUBSYSTEM USING A SET OF ADDRESSES THAT TEST THE ADAPTER/DEVICE INTERFACE AND ELECTROMECHANICAL HEAD POSITIONING HARDWARE.

PROCEDURE:

EACH TEST WILL RECALIBRATE THE DRIVE IF "PIP" OR "SKI" IS ACTIVE. FOLLOWING THAT, THE TEST EXECUTES A SEEK TO A CYLINDER ADDRESS OR SERIES OF ADDRESSES. AT THE COMPLETION OF EACH SEEK OPERATION, SUBSYSTEM STATUS IS STORED AND THE TEST CHECKS FOR PRIMARY ERRORS WHICH PRECLUDE FURTHER ERROR CHECKING. IF THERE ARE NO PRIMARY ERRORS THE TEST CHECKS FOR OPERATIONAL ERRORS AND THEN SECONDARY ERRORS.

SEEK TO LAST CYLINDER TEST

THIS TEST SEEKS TO THE LAST CYLINDER, I.E., CYLINDER 822.

SEEK TO FIRST CYLINDER TEST

THIS TEST SEEKS TO THE FIRST CYLINDER, I.E. CYLINDER 0.

SEEK PRIME CYLINDERS TEST

THIS TEST SEEKS FORWARD TO EACH PRIME CYLINDER ADDRESS, I.E, CYLINDERS 1,2,4,8,...,512.

SEEK ZERO DIFFERENCE TEST

THIS TEST EXECUTES SUCCESSIVE SEEKS TO CYLINDER 0.

SEEK MAXIMUM DIFFERENCE FORWARD TEST

THIS TEST SEEKS TO CYLINDER 0 FOLLOWED BY A SEEK TO THE LAST CYLINDER.

SEEK ADJACENT FORWARD TEST

THIS TEST SEEKS TO CYLINDER 0, FOLLOWED BY A SEEK TO THE ADJACENT FORWARD CYLINDER, I.E., CYLINDER 1.

SEEK ADJACENT REVERSE TEST

THIS TEST SEEKS TO CYLINDER 1, FOLLOWED BY A SEEK TO THE ADJACENT REVERSE CYLINDER, I.E., CYLINDER 0.

SEEK TO INVALID SECTOR TEST

WITH THE OFFSET REGISTER SET FOR 18 BIT FORMAT, THE PROGRAM SEEKS TO CYLINDER 0, TRACK 0, FOR EACH INVALID SECTOR ADDRESS AND VERIFIES THAT INVALID ADDRESS ERROR SETS FOR EACH SECTOR.

SEEK TO INVALID TRACK TEST

THE TEST SEEKS TO EACH INVALID TRACK ADDRESS WITH CYLINDER AND SECTOR ADDRESS SET TO 0 AND VERIFIES THAT INVALID ADDRESS ERROR SETS FOR EACH TRACK ADDRESS.

SEEK TO INVALID CYLINDER TEST

THE PROGRAM SEEKS TO EACH INVALID CYLINDER ADDRESS WITH THE SECTOR AND TRACK ADDRESS SET TO 0 AND VERIFIES THAT INVALID ADDRESS ERROR SETS FOR EACH CYLINDER ADDRESSS.

IVC SEEK TEST

PURPOSE:

TO VERIFY THAT INVALID COMMAND STATUS, B'T 12 OF RMER2 SETS WHEN VOLUME VALID IS INACTIVE DURING A SEEK COMMAND.

PROCEDURE:

THE TEST RESETS VOLUME VALID BY SETTING AND RESETTING DIAGNOSTIC MODE, THEN EXECUTES A SEEK COMMAND AND VERIFIES THAT "IVC" STATUS IS SET.

ABORT SEEK TEST

PURPOSE:

TO VERIFY THAT THE RM05/3/2 INHIBITS A SEEK WHEN AN ABORT CONDITION EXISTS AT THE START OF THE COMMAND.

PROCEDURE:

THE TEST SETS AN ERROR IN THE ERROR REGISTER AND ISSUES A SEEK COMMAND, VERIFYING THAT "PIP" REMAINS INACTIVE.

SEEK AT OFFSET

PURPOSE:

TO VERIFY THAT OFFSET MODE DOES NOT CAUSE ERRORS DURING SEEK.

PROCEDURE:

THE TEST EXECUTES AN OFFSET COMMAND THEN EXECUTES A SEEK COMMAND, VERIFYING THE RESULTS OF THE SEEK.

LOOK AHEAD TEST

PURPOSE:

TO INSURE THAT THE SECTOR COUNT WHICH ORICINATES AT THE DRIVE AND IS VISIBLE THROUGH THE LOOK AHEAD REGISTER (RMLA) IS OPERATIONAL.

PROCEDURE:

WITH THE OFFSET REGISTER SET FOR 18 BIT FORMAT, THE PROGRAM SAMPLES THE LOOK AHEAD REGISTER AND COLLECTS EACH DIFFERENT SAMPLE UNTIL THE VALUE OF THE FIRST SAMPLE IS DETECTED OR UNTIL 31. SAMPLES ARE TAKEN. THE COLLECTION IS THEN TESTED TO DETERMINE THAT THE SECTOR COUNT INCREMENTS CORRECTLY THROUGH THE ENTIRE RANGE OF VALID SECTORS. THE SAME PROCEDURE IS REPEATED FOR 16 BIT FORMAT, WHERE THE LIMIT ON THE NUMBER OF SAMPLES IS 33.

SEARCH EACH SECTOR ON CYLINDER

PURPOSE:

TO VERIFY THE EXECUTION OF SEARCH OPERATIONS WITH NO HEAD MOTION USING THE SECTOR PULSE FOR SECTOR COMPARE.

PROCEDURE:

THE TEST INTIALIZES AND RECALIBRATES THE DRIVE IF "PIP" OR "SKI" IS ACTIVE THEN SEEKS TO CYLINDER 0. THE TEST THEN DOES A SEARCH TO EACH SECTOR AND VERIFIES THAT THE SEARCH COMPLETES WITHOUT ERROR. THIS TEST IS DONE ONCE IN 18 BIT FORMAT AND ONCE IN 16 BIT FORMAT.

SEARCH OFF CYLINDER

PURPOSE:

TO VERIFY THE EXECUTION OF A SEARCH COMMAND WITH IMPLIED HEAD MOTION.

PROCEDURE:

THE DRIVE IS RECALIBRATED AND INITIALIZED IF THE DRIVE IS

OFF CYLINDER OR HAS A SEEK ERROR. THE PROGRAM THEN EXECUTES AND EXPLICIT SEEK TO CYLINDER 410., FOLLOWED BY A SEARCH TO CYLINDER 411., TRACK 0, SECTOR 0. THE FOLLOWING IS SEARCHED FOR BY AN EXPLICIT SEEK TO CYLINDER 409., FOLLOWED BY A SEARCH TO CYLINDER 412., TRACK 0, SECTOR 1, ETC., UNTIL ALL THE SECTORS HAVE BEEN SEARCHED. THE TEST IS DONE FOR 18 BIT FORMAT, WHERE THE LAST SECTOR SEARCHED IS SECTOR 29. THE TEST IS REPEATED FOR 16 BIT FORMAT, WHERE THE LAST SECTOR SEARCHED IS 31.

SEARCH INVALID SECTOR

PURPOSE:

TO VERIFY THE DETECTION OF AN INVALID SECTOR ADDRESS DURING A SEARCH OPERATION.

PROCEDURE:

THE DRIVE IS RECALIBRATED AND INITIALIZED IF IT IS OFF CYLINDER OR HAS A SEEK ERROR. WITH THE OFFSET REGISTER SET FOR 18 BIT FORMAT, THE PROGRAM EXECUTES A SEARCH TO EACH INVALID SECTOR, I.E., SECTORS 30 AND 31 AND VERIFIES THAT INVALID ADDRESS ERROR SETS FOR EACH SECTOR.

SEARCH INVALID TRACK

PURPOSE:

TO VERIFY THE DETECTION OF AN INVALID TRACK ADDRESS DURING A SEARCH OPERATION.

PROCEDURE:

THE DRIVE IS RECALIBRATED AND INITIALIZED IF IT IS OFF CYLINDER OR HAS A SEEK ERROR. THE PROGRAM EXECUTES A SEARCH TO EACH INVALID TRACK ADDRESS WITH THE CYLINDER ADDRESS 0, AND SECTOR ADDRESS 0 AND VERIFIES THAT INVALID ADDRESS ERROR SETS FOR EACH TRACK.

SEARCH INVALID CYLINDER

PURPOSE:

TO VERIFY THE DETECTION OF AN INVALID CYLINDER ADDRESS DURING A SEARCH OPERATION.

```
1255
1256                                    PROCEDURE:
1257
1258                                        THE DRIVE IS RECALIBRATED AND INITIALIZED IF IT IS OFF
1259                                    CYLINDER OR HAS A SEEK ERROR.  THE PROGRAM THEN EXECUTES A SEARCH
1260                                    TO EACH INVALID CYLINDER ADDRESS AND VERIFIES THAT INVALID
1261                                    ADDRESS ERROR SETS FOR EACH CYLINDER.
1262
1263
1264
1265
1266
1267                                IVC SEARCH TEST
1268
1269                                    PURPOSE:
1270
1271                                        TO VERIFY THAT INVALID COMMAND STATUS SETS WHEN VOLUME VALID
1272                                    IS INACTIVE DURING A SEARCH COMMAND.
1273
1274                                    PROCEDURE:
1275
1276                                        VOLUME VALID IS RESET BY SETTING AND RESETTING DIAGNOSTIC
1277                                    MODE,  AFTER WHICH THE TEST EXECUTES A SEARCH COMMAND, VERIFYING
1278                                    THAT "IVC" STATUS SETS.
1279
1280
1281
1282
1283
1284                                ABORT SEARCH TEST
1285
1286                                    PURPOSE:
1287
1288                                        TO VERIFY THAT THE RM05/3/2 INHIBITS A SEARCH WHEN AN  ABORT
1289                                    CONDITION EXISTS AT THE START OF THE COMMAND.
1290
1291                                    PROCEDURE:
1292
1293                                        THE TEST SETS AN ERROR IN THE ERROR REGISTER AND ISSUES A
1294                                    SEARCH COMMAND, VERIFYING THAT "PIP" REMAINS INACTIVE.
1295
1296
1297
1298
1299
1300                                SEARCH AT OFFSET TEST
1301
1302                                    PURPOSE:
1303
1304                                        TO VERIFY THAT OFFSET MODE DOES NOT CAUSE SEARCH ERRORS.
1305
1306                                    PROCEDURE:
1307
1308                                        THE TEST EXECUTES AN OFFSET COMMAND, THEN EXECUTES A SEARCH
1309                                    COMMAND, VERIFYING THAT THERE ARE NO ERRORS DURING THE SEARCH.
1310
1311
```

```
1312
1313
1314
1315                                    \
```

```
    1                       ;PROGRAM REVISION #001
478
479                         .TITLE  CZRMMA0 RM05/3/2 FCTNL TST 1
                            ;*COPYRIGHT (C) 1980
                            ;*DIGITAL EQUIPMENT CORP.
                            ;*MAYNARD, MASS. 01754
                            ;*
                            ;*PROGRAM BY MIKE LEAVITT
                            ;*
                            ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                            ;*PACKAGE (MAINDEC-11-DZQAC-C4), 1980.
                            ;*
480                         .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>
481                         ;*         7            TN128
                            ;*         6            TN64
                            ;*         5            TN32
                            ;*         4            TN16
                            ;*         3            TN8
                            ;*         2            TN4
                            ;*         1            TN2
                            ;*         0            TN1
482
483                         .SBTTL  BASIC DEFINITIONS

                            ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                      STACK   = 1100
104000                      ERROR   = EMT           ;;BASIC DEFINITION OF ERROR CALL
000004                      SCOPE   = IOT           ;;BASIC DEFINITION OF SCOPE CALL

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

                            ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                      R0      = %0            ;;GENERAL REGISTER
000001                      R1      = %1            ;;GENERAL REGISTER
000002                      R2      = %2            ;;GENERAL REGISTER
000003                      R3      = %3            ;;GENERAL REGISTER
```

```
000004                  R4      = %4            ;;GENERAL REGISTER
000005                  R5      = %5            ;;GENERAL REGISTER
000006                  R6      = %6            ;;GENERAL REGISTER
000007                  R7      = %7            ;;GENERAL REGISTER
000006                  SP      = %6            ;;STACK POINTER
000007                  PC      = %7            ;;PROGRAM COUNTER

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

                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                  SW15    = 100000
040000                  SW14    = 40000
020000                  SW13    = 20000
010000                  SW12    = 10000
004000                  SW11    = 4000
002000                  SW10    = 2000
001000                  SW09    = 1000
000400                  SW08    = 400
000200                  SW07    = 200
000100                  SW06    = 100
000040                  SW05    = 40
000020                  SW04    = 20
000010                  SW03    = 10
000004                  SW02    = 4
000002                  SW01    = 2
000001                  SW00    = 1
001000                  SW9=SW09
000400                  SW8=SW08
000200                  SW7=SW07
000100                  SW6=SW06
000040                  SW5=SW05
000020                  SW4=SW04
000010                  SW3=SW03
000004                  SW2=SW02
000002                  SW1=SW01
000001                  SW0=SW00

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15   = 100000
040000                  BIT14   = 40000
020000                  BIT13   = 20000
010000                  BIT12   = 10000
004000                  BIT11   = 4000
002000                  BIT10   = 2000
001000                  BIT09   = 1000
000400                  BIT08   = 400
000200                  BIT07   = 200
000100                  BIT06   = 100
000040                  BIT05   = 40
```

```
                000020          BIT04   = 20
                000010          BIT03   = 10
                000004          BIT02   = 4
                000002          BIT01   = 2
                000001          BIT00   = 1
                001000          BIT9=BIT09
                000400          BIT8=BIT08
                000200          BIT7=BIT07
                000100          BIT6=BIT06
                000040          BIT5=BIT05
                000020          BIT4=BIT04
                000010          BIT3=BIT03
                000004          BIT2=BIT02
                000002          BIT1=BIT01
                000001          BIT0=BIT00

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                000004          ERRVEC  = 4             ;;TIME OUT AND OTHER ERRORS
                000010          RESVEC  = 10            ;;RESERVED AND ILLEGAL INSTRUCTIONS
                000014          TBITVEC = 14            ;;"T" BIT
                000014          TRTVEC  = 14            ;;TRACE TRAP
                000014          BPTVEC  = 14            ;;BREAKPOINT TRAP (BPT)
                000020          IOTVEC  = 20            ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                000024          PWRVEC  = 24            ;;POWER FAIL
                000030          EMTVEC  = 30            ;;EMULATOR TRAP (EMT) **ERROR**
                000034          TRAPVEC = 34            ;;"TRAP" TRAP
                000060          TKVEC   = 60            ;;TTY KEYBOARD VECTOR
                000064          TPVEC   = 64            ;;TTY PRINTER VECTOR
                000240          PIRQVEC = 240           ;;PROGRAM INTERRUPT REQUEST VECTOR
484
485                             .SBTTL  RM REGISTER BIT DEFINITIONS
486
487                             ;*RMCS1 CONTROL STATUS REGISTER
488
489             004000          DVA     = BIT11                 ;DEVICE AVAILABLE-READ ONLY
490             000040          F4      = BIT05                 ;FUNCTION CODE
491             000020          F3      = BIT04                 ;FUNCTION CODE
492             000010          F2      = BIT03                 ;FUNCTION CODE
493             000004          F1      = BIT02                 ;FUNCTION CODE
494             000002          F0      = BIT01                 ;FUNCTION CODE
495             000001          GO      = BIT00                 ;GO BIT
496             000077          FNCMSK  = 000077                ;FUNCTION CODE MASK
497
498                             ;FUNCTION CODES (BITS 01-05 OF RMCS1)
499             000000          NOP     = 000000                ;NOP COMMAND
500             000002          ILF02   = 000002                ;ILLEGAL COMMAND
501             000004          SEEK    = 000004                ;SEEK COMMAND
502             000006          RECAL   = 000006                ;RECALIBRATE COMMAND
503             000010          DRVCLR  = 000010                ;DRIVE CLEAR COMMAND
504             000012          RLEASE  = 000012                ;RELEASE COMMAND
505             000014          OFFSET  = 000014                ;OFFSET COMMAND
506             000016          RTC     = 000016                ;RETURN TO CENTERLINE COMMAND
507             000020          RIP     = 000020                ;READ IN PRESET COMMAND
508             000022          PAKACK  = 000022                ;PACK ACKNOWLEDGE COMMAND
509             000022          PACACK  = PAKACK
510             000024          ILF24   = 000024                ;ILLEGAL COMMAND
511             000026          ILF26   = 000026                ;ILLEGAL COMMAND
```

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

                        ;*RMDA  DISK ADDRESS REGISTER

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

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

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

                        ;*RMDS  DRIVE STATUS REGISTER

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

                        ;*RMER1 ERROR REGISTER #1
```

```
                        
000000                  DCK     = BIT15                 ;DATA CHECK ERROR
000000                  UNS     = BIT14                 ;DRIVE UNSAFE
000000                  OPI     = BIT13                 ;OPERATION INCOMPLETE
000000                  DTE     = BIT12                 ;DRIVE TIMING ERROR
000000                  WLE     = BIT11                 ;WRITE LOCK ERROR
000000                  IAE     = BIT10                 ;INVALID ADDRESS ERROR
000000                  AOE     = BIT09                 ;ADDRESS OVERFLOW ERROR
000000                  HCRC    = BIT08                 ;HEADER CRC ERROR
000000                  HCE     = BIT07                 ;HEADER COMPARE ERROR
000000                  ECH     = BIT06                 ;ECC "HARD" ERROR
000000                  WCF     = BIT05                 ;WRITE CLOCK FAILURE
000000                  FER     = BIT04                 ;FORMAT ERROR
000000                  PAR     = BIT03                 ;PARITY ERROR
000000                  RMR     = BIT02                 ;REGISTER MODIFICATION REFUSED
000000                  ILR     = BIT01                 ;ILLEGAL REGISTER
000000                  ILF     = BIT00                 ;ILLEGAL FUNCTION
                        
115760                  NDTMSK  = DCK!DTE.WLE!AOE.HCRC!HCE!ECH!WCF!FER
                        ;"NDTMSK" IS USED TO MASK ERROR REGISTER 1 DURING NON - DATA
                        ;COMMANDS, I.E., HOUSEKEEPING AND POSITIONING COMMANDS
                        
                        ;*RMAS  ATTENTION SUMMARY REGISTER
                        
000377                  ATNMSK  = 377                   ;MASK FOR ATTENTION BITS
                        
                        ;*RMLA  LOOK AHEAD REGISTER
                        
002000                  SC4     = BIT10                 ;SECTOR COUNT = 16
001000                  SC3     = BIT09                 ;SECTOR COUNT = 8
000400                  SC2     = BIT08                 ;SECTOR COUNT = 4
000200                  SC1     = BIT07                 ;SECTOR COUNT = 2
000100                  SC0     = BIT06                 ;SECTOR COUNT = 1
                        
003700                  SCTMSK  = 003700                ;SECTOR COUNT MASK
                        
                        ;*RMMR1 MAINTENANCE REGISTER #1
                        
                        ;WRITE ONLY BITS
100000                  DBCK    = BIT15                 ;DEBUG CLOCK
040000                  DBEN    = BIT14                 ;DEBUG CLOCK ENABLE
020000                  DEBL    = BIT13                 ;DIAGNOSTIC END OF BLOCK
010000                  DTO     = BIT12                 ;DIAGNOSTIC TIMEOUT
004000                  MCLK    = BIT11                 ;MAINTENANCE CLOCK
002000                  MRD     = BIT10                 ;READ DATA
001000                  MUR     = BIT09                 ;UNIT READY
000400                  MOC     = BIT08                 ;ON CYLINDER
000200                  MSER    = BIT07                 ;SEEK ERROR
000100                  MDF     = BIT06                 ;DRIVE FAULT
000040                  MS      = BIT05                 ;SECTOR PULSE
000010                  MWP     = BIT03                 ;WRITE PROTECT
000004                  MI      = BIT02                 ;INDEX PULSE
000002                  MSC     = BIT01                 ;SECTOR COMPARE
000001                  DMD     = BIT00                 ;DIAGNOSTIC MODE
                        
                        ;READ ONLY BITS
100000                  OCC     = BIT15                 ;OCCUPIED
```

```
621          040000              RG      = BIT14                 ;RUN AND GO
622          020000              EBL     = BIT13                 ;END OF BLOCK
623          010000              REX     = BIT12                 ;EXCEPTION
624          004000              ESRC    = BIT11                 ;ENABLE SEARCH
625          002000              PLFS    = BIT10                 ;LOOKING FOR SYNC
626          001000              ECRC    = BIT09                 ;ENABLE CRC OUT
627          000400              PDA     = BIT08                 ;DATA AREA
628          000200              PHA     = BIT07                 ;HEADER AREA
629          000100              CONT    = BIT06                 ;CONTINUE
630          000040              WC      = BIT05                 ;WORD CLOCK
631          000020              EECC    = BIT04                 ;ENABLE ECC OUT
632          000010              MWD     = BIT03                 ;WRITE DATA BIT
633          000004              LS      = BIT02                 ;LAST SECTOR
634          000002              LST     = BIT01                 ;LAST SECTOR AND TRACK
635          000001              DMD     = BIT00                 ;DIAGNOSTIC MODE
636
637                              ;*RMDT  DRIVE TYPE REGISTER
638
639          100000              NSA     = BIT15                 ;NOT SECTOR ADDRESSED - 0
640          040000              TAP     = BIT14                 ;TAPE DRIVE = 0
641          020000              MOH     = BIT13                 ;MOVING HEAD = 1
642          004000              DRQ     = BIT11                 ;DRIVE REQUEST REQUIRED
643
644          020024              SNGPRT  = 020024                ;SINGLE PORT DRIVE TYPE (RM02)
645          024024              DULPRT  = 024024                ;DUAL PORT DRIVE TYPE (RM02)
646
647                              ;*RMOF  OFFSET REGISTER
648
649          010000              FMT16   = BIT12                 ;16 BIT WORD FORMAT
650          004000              ECI     = BIT11                 ;ECC INHIBIT
651          002000              HCI     = BIT10                 ;HEADER COMPARE INHIBIT
652          000200              OFD     = BIT07                 ;OFFSET FORWARD
653
654                              ;*RMDC  DESIRED CYLINDER ADDRESS REGISTER
655
656          001777              CYLMSK  = 001777                ;MASK FOR CYLINDER ADDRESS
657
658                              ;*RMMR2 MAINTENANCE REGISTER #2
659
660                              ;READ ONLY BITS
661          100000              RQA     = BIT15                 ;PORT A REQUEST
662          040000              RQB     = BIT14                 ;PORT B REQUEST
663          020000              TAG     = BIT13                 ;TAG CONTROL
664          010000              TST     = BIT12                 ;COMMAND SEQUENCE TEST BIT
665          004000              CC      = BIT11                 ;CONTROL OR CYLINDER TAG
666          002000              CH      = BIT10                 ;CONTROL OR HEAD TAG
669          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
670
```

```
                                ;*RMER2 ERROR REGISTER 2

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

                .SBTTL  PROGRAM MNEMONICS

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

                .SBTTL  RM REGISTER INDEX VALUES

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

000077          IDXMSK  = 77                    ;MASK FOR REGISTER INDEX NUMBER

                .SBTTL  RH CONTROLLER REGISTER BIT DEFINITIONS

                ;*RMCS1 CONTROL STATUS REGISTER #1

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

```
719          001000                  A17     = BIT09                 ;ADDRESS EXTENSION
720          000400                  A16     = BIT08                 ;ADDRESS EXTENSION
721          000200                  RDY     = BIT07                 ;READY-READ ONLY
722          000100                  IE      = BIT06                 ;INTERRUPT ENABLE
723
724                                  ;*RMCS2 RH CONTROL STATUS REGISTER #2
725
726          100000                  DLT     = BIT15                 ;DATA LATE-READ ONLY
727          040000                  WCE     = BIT14                 ;WRITE CHECK ERROR-READ ONLY
728          020000                  UPE     = BIT13                 ;UNIBUS PARITY ERROR
729          010000                  NED     = BIT12                 ;NONEXISTANT DRIVE-READ ONLY
730          004000                  NEM     = BIT11                 ;NONEXISTANT MEMORY-READ ONLY
731          002000                  PGE     = BIT10                 ;PROGRAM ERROR-READ ONLY
732          001000                  MXF     = BIT09                 ;MISSED TRANSFER
733          000400                  MDPE    = BIT08                 ;MASSBUS DATA BUS PARITY ERROR-READ ONLY
734          000200                  OR      = BIT07                 ;OUTPUT READY-READ ONLY
735          000100                  IR      = BIT06                 ;INPUT READY-READ ONLY
736          000040                  CLR     = BIT05                 ;CONTROLLER CLEAR
737          000020                  PAT     = BIT04                 ;PARITY TEST
738          000010                  BAI     = BIT03                 ;UNIBUS ADDRESS INCREMENT INHIBIT
741          000004                  U2      = BIT02                 ;UNIT SELECT
             000002                  U1      = BIT01                 ;UNIT SELECT
             000001                  U0      = BIT00                 ;UNIT SELECT
742
743                                  ;UNIT SELECT MASK
744
745          000007                  UNTMSK  = 7                     ;UNIT SELECT MASK
746
747                                  ;*RMCS3 RH70 CONTROL STATUS REGISTER #3
748
749          100000                  APE     = BIT15                 ;ADDRESS PARITY ERROR
750          040000                  DPEHI   = BIT14                 ;DATA PARITY ERROR HIGH WORD
751          020000                  DPELO   = BIT13                 ;DATA PARITY ERROR LOW WORD
752          010000                  WCEHI   = BIT12                 ;WRITE CHECK ERROR HIGH WORD
753          004000                  WCELO   = BIT11                 ;WRITE CHECK ERROR LOW WORD
754          002000                  DBL     = BIT10                 ;DOUBLE WORD TRANSFER
755          000100                  IE      = BIT06                 ;INTERRUPT ENABLE
756          000010                  IPCK3   = BIT03                 ;INVERT PARITY CHECK
757          000004                  IPCK2   = BIT02                 ;INVERT PARITY CHECK
758          000002                  IPCK1   = BIT01                 ;INVERT PARITY CHECK
759          000001                  IPCK0   = BIT00                 ;INVERT PARITY CHECK
760
761                                  .SBTTL  RH11/RH70 CONTROLLER REGISTER INDEX VALUES
762
763          000000                  RMCS1   = 00                    ;CONTROL, STATUS REGISTER #1
764          000002                  RMWC    = 02                    ;WORD COUNT REGISTER
765          000004                  RMBA    = 04                    ;BUS ADDRESS REGISTER
766          000010                  RMCS2   = 10                    ;CONTROL, STATUS REGISTER #2
767          000022                  RMDB    = 22                    ;DATA BUFFER
768          000050                  RMBAE   = 50                    ;BUS ADDRESS EXTENSION
769          000052                  RMCS3   = 52                    ;CONTROL, STATUS REGISTER #3
770
771          176700                  ABASE   = 176700                ;UNIBUS ADDRESS
772          120254                  AVECT1  = 120254                ;UNIBUS VECTOR ADDRESS AND PRIORITY
773
775
776                                  .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  005412                  JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
777
778                                     .SBTTL  ACT11 HOOKS

                                        ;;**************************************************************
                                        ;HOOKS REQUIRED BY ACT11
                000204                          $SVPC=.                 ;SAVE PC
                000046                          .=46
     000046     042326                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                          .=52
     000052     000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                000204                          .=$SVPC                 ;; RESTORE PC
779
780             001100                  .=1100
781                                     .SBTTL  APT PARAMETER BLOCK

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

     001100                             $APTHD:
     001100     000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
     001102     001222                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
     001104     000001                  $TSTM:  .WORD   1       ;;RUN TIM OF LONGEST TEST
     001106     000002                  $PASTM: .WORD   2       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
     001110     000002                  $UNITM: .WORD   2       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDED UNIT
     001112     000042                          .WORD   $ETEND-$MAIL/2  ;;LENGTH MAILBOX-ETABLE(WORDS)
782             001114                  TAGADR=.
```

```
      0                         .SBTTL  COMMON TAGS

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

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

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

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

```
0                               .SBTTL  USER DEFINED TAGS

001326  000000                  CTLFG:  .WORD   0       ;CONTAINS CONTROL-C FLAG
001330  000000                  XXDP:   .WORD   0       ;THE LOW BYTE CONTAINS THE DRIVE NUMBER FROM WHICH
                                                        ;THE PROGRAM WAS LOADED. THE HIGH BYTE CONTAINS THE
                                                        ;'XXDP' DEVICE CODE FOR THE RM05/3/2.
001332  000                     LSTRK:  .BYTE   0       ;LO BYTE = 0
001333  000                             .BYTE   0       ;HI BYTE, CONTAINS LAST TRACK ADDRESS OF UNIT
                                                        ;UNDER TEST. RM02/3 = 4., RM05 = 18.

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

001334                          GETBUF:

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

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

001410                          PUTBUF:

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

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

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

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

001510  000000          CLKADR: .WORD   0               ;UNIBUS ADDRESS OF KW11 CLOCK
001512  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.
001514                  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.
001543                  PUTINX: .BLKB   23.             ;PUT INDEX TABLE

                        ;PUT TAGS HERE
```

```
      0                                 .SBTTL  ERROR POINTER TABLE

                                        ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                        ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                        ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                        ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                        ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                        ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                        ;*      DH              ;;POINTS TO THE DATA HEADER
                                        ;*      DT              ;;POINTS TO THE DATA
                                        ;*      DF              ;;POINTS TO THE DATA FORMAT


    001572                              $ERRTB:
  1
  2
  3                                     ;ERROR  1       WRONG UNIT SELECTED
  4
    001572  067604                              EMT1
    001574  073676                              EHT1
    001576  074022                              EDT1
    001600  074112                              EFT1

  5
  6                                     ;ERROR  2       DEVICE WENT UNAVAILABLE
  7
    001602  067610                              EMT2
    001604  073676                              EHT1
    001606  074022                              EDT1
    001610  074112                              EFT1

  8
  9                                     ;ERROR  3       DEVICE WENT NONEXISTENT
 10
    001612  067616                              EMT3
    001614  073676                              EHT1
    001616  074022                              EDT1
    001620  074112                              EFT1

 11
 12                                     ;ERROR  4       CONTROLLER NOT READY
 13
    001622  067624                              EMT4
    001624  073676                              EHT1
    001626  074022                              EDT1
    001630  074112                              EFT1

 14
 15                                     ;ERROR  5       DRIVE NOT READY AND GO NOT RESET
 16
    001632  067632                              EMT5
    001634  073676                              EHT1
    001636  074022                              EDT1
    001640  074112                              EFT1

 17
```

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

001642  067640                          EMT6
001644  073676                          EHT1
001646  074022                          EDT1
001650  074112                          EFT1

                                ;ERROR  7       BUS TIMEOUT TRYING TO READ OR WRITE REGISTER

001652  067646                          EMT7
001654  000000                          0
001656  000000                          0
001660  000000                          0

                                ;ERROR  10      DRIVE NOT READY BUT GO IS RESET

001662  067654                          EMT10
001664  073676                          EHT1
001666  074022                          EDT1
001670  074112                          EFT1

                                ;ERROR  11      GO NOT RESET BUT DRIVE IS READY

001672  067660                          EMT11
001674  073676                          EHT1
001676  074022                          EDT1
001700  074112                          EFT1

                                ;ERROR  12      INCORRECT FUNCTION CODE

001702  067664                          EMT12
001704  073676                          EHT1
001706  074022                          EDT1
001710  074112                          EFT1

                                ;ERROR  13      PARITY ERROR READING REMOTE REGISTERS

001712  067672                          EMT13
001714  073676                          EHT1
001716  074022                          EDT1
001720  074112                          EFT1

                                ;ERROR  14      TRANSFER ERROR IS INCORRECT

001722  067704                          EMT14
001724  073676                          EHT1
001726  074022                          EDT1
001730  074112                          EFT1

                                ;ERROR  15      INCORRECT WORD COUNT
```

```
40
   001732  067712                          EMT15
   001734  073676                          EHT1
   001736  074022                          EDT1
   001740  074112                          EFT1

41
42                                 ;ERROR  16      INCORRECT BUS ADDRESS
43
   001742  067720                          EMT16
   001744  073676                          EHT1
   001746  074022                          EDT1
   001750  074112                          EFT1

44
45                                 ;ERROR  17      INCORRECT LBT STATUS
46
   001752  067730                          EMT17
   001754  073676                          EHT1
   001756  074022                          EDT1
   001760  074112                          EFT1

47
48                                 ;ERROR  20      INCORRECT AOE
49
   001762  067740                          EMT20
   001764  073676                          EHT1
   001766  074022                          EDT1
   001770  074112                          EFT1

50
51                                 ;ERROR  21      INCORRECT DISK ADDRESS
52
   001772  067750                          EMT21
   001774  073676                          EHT1
   001776  074022                          EDT1
   002000  074112                          EFT1

53
54                                 ;ERROR  22      INCORRECT CYLINDER ADDRESS
55
   002002  067760                          EMT22
   002004  073676                          EHT1
   002006  074022                          EDT1
   002010  074112                          EFT1

56
57                                 ;ERROR  23      INCORRECT WLE STATUS
58
   002012  067770                          EMT23
   002014  073676                          EHT1
   002016  074022                          EDT1
   002020  074112                          EFT1

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

```
002022  070000                          EMT24
002024  073676                          EHT1
002026  074022                          EDT1
002030  074112                          EFT1

62
63                              ;ERROR  25      INCORRECT WCF STATUS
64
002032  070010                          EMT25
002034  073676                          EHT1
002036  074022                          EDT1
002040  074112                          EFT1

65
66                              ;ERROR  26      INCORRECT WCE STATUS
67
002042  070020                          EMT26
002044  073676                          EHT1
002046  074022                          EDT1
002050  074112                          EFT1

68
69                              ;ERROR  27      INCORRECT MDPE STATUS
70
002052  070030                          EMT27
002054  073676                          EHT1
002056  074022                          EDT1
002060  074112                          EFT1

71
72                              ;ERROR  30      INCORRECT DCK STATUS
73
002062  070040                          EMT30
002064  073676                          EHT1
002066  074022                          EDT1
002070  074112                          EFT1

74
75                              ;ERROR  31      INCORRECT ECH STATUS
76
002072  070050                          EMT31
002074  073676                          EHT1
002076  074022                          EDT1
002100  074112                          EFT1

77
78                              ;ERROR  32      DLT SHOULD NOT BE SET
79
002102  070060                          EMT32
002104  073676                          EHT1
002106  074022                          EDT1
002110  074112                          EFT1

80
81                              ;ERROR  33      MXF SHOULD NOT BE SET
82
002112  070070                          EMT33
```

```
                002114  073676                          EHT1
                002116  074022                          EDT1
                002120  074112                          EFT1

     83
     84                                 ;ERROR  34       DTE SHOULD NOT BE SET
     85
                002122  070100                          EMT34
                002124  073676                          EHT1
                002126  074022                          EDT1
                002130  074112                          EFT1

     86
     87                                 ;ERROR  35       INCORRECT HCRC STATUS
     88
                002132  070110                          EMT35
                002134  073676                          EHT1
                002136  074022                          EDT1
                002140  074112                          EFT1

     89
     90                                 ;ERROR  36       INCORRECT HCE STATUS
     91
                002142  070120                          EMT36
                002144  073676                          EHT1
                002146  074022                          EDT1
                002150  074112                          EFT1

     92
     93                                 ;ERROR  37       INCORRECT FER STATUS
     94
                002152  070130                          EMT37
                002154  073676                          EHT1
                002156  074022                          EDT1
                002160  074112                          EFT1

     95
     96                                 ;ERROR  40       DPE SHOULD NOT BE SET (NOT A DATA COMMAND)
     97
                002162  070140                          EMT40
                002164  073676                          EHT1
                002166  074022                          EDT1
                002170  074112                          EFT1

     98
     99                                 ;ERROR  41       LOST "MOL" DURING PACK ACKNOWLEDGE
    100
                002172  070146                          EMT41
                002174  073676                          EHT1
                002176  074022                          EDT1
                002200  074112                          EFT1

    101
    102                                 ;ERROR  42       UNSAFE ERROR DURING PACK ACKNOWLEDGE
    103
                002202  070156                          EMT42
                002204  073676                          EHT1
```

```
002206  074022                          EDT1
002210  074112                          EFT1

104
105                             ;ERROR  43      "OPI" ERROR DURING PACK ACKNOWLEDGE
106
002212  070170                          EMT43
002214  073676                          EHT1
002216  074022                          EDT1
002220  074112                          EFT1

107
108                             ;ERROR  44      "RMR" ERROR DURING PACK ACKNOWLEDGE
109
002222  070200                          EMT44
002224  073676                          EHT1
002226  074022                          EDT1
002230  074112                          EFT1

110
111                             ;ERROR  45      "ILR" ERROR DURING PACK ACKNOWLEDGE
112
002232  070210                          EMT45
002234  073676                          EHT1
002236  074022                          EDT1
002240  074112                          EFT1

113
114                             ;ERROR  46      "ILF" ERROR DURING PACK ACKNOWLEDGE
115
002242  070220                          EMT46
002244  073676                          EHT1
002246  074022                          EDT1
002250  074112                          EFT1

116
117                             ;ERROR  47      COMPOSITE ERROR STATUS IS INCORRECT
118
002252  070230                          EMT47
002254  073676                          EHT1
002256  074022                          EDT1
002260  074112                          EFT1

119
120                             ;ERROR  50      PARITY ERROR WRITING REMOTE REGISTERS
121
002262  070236                          EMT50
002264  073676                          EHT1
002266  074022                          EDT1
002270  074112                          EFT1

122
123                             ;ERROR  51      INCORRECT IAE STATUS DURING SEEK COMMAND
124
002272  070246                          EMT51
002274  073676                          EHT1
002276  074022                          EDT1
```

```
002300  074112                          EFT1

125
126                             ;ERROR  52      OPI ERROR DURING SEEK - MEDIUM IS NOT ON LINE
127
002302  070260                          EMT52
002304  073676                          EHT1
002306  074022                          EDT1
002310  074112                          EFT1

128
129                             ;ERROR  53      OPI ERROR DURING SEEK - MEDIUM IS ON LINE, ASSUME
130                             ;               ON CYLINDER LATCH DIDN'T RESET
131
002312  070276                          EMT53
002314  073676                          EHT1
002316  074022                          EDT1
002320  074112                          EFT1

132
133                             ;ERROR  54      SEEK INCOMPLETE ERROR DURING SEEK COMMAND
134
002322  070314                          EMT54
002324  073676                          EHT1
002326  074022                          EDT1
002330  074112                          EFT1

135
136                             ;ERROR  55      DEVICE CHECK DURING SEEK COMMAND
137
002332  070324                          EMT55
002334  073676                          EHT1
002336  074022                          EDT1
002340  074112                          EFT1

138
139                             ;ERROR  56      PIP IS STILL SET AFTER SEEK - SKI IS RESET
140
002342  070336                          EMT56
002344  073676                          EHT1
002346  074022                          EDT1
002350  074112                          EFT1

141
142                             ;ERROR  57      ATA DID NOT SET DURING SEEK COMMAND
143
002352  070354                          EMT57
002354  073676                          EHT1
002356  074022                          EDT1
002360  074112                          EFT1

144
145                             ;ERROR  60      IVC ERROR DURING SEEK COMMAND - LOST
146                             ;               VOLUME VALID
147
002362  070364                          EMT60
002364  073676                          EHT1
```

```
002366  074022                          EDT1
002370  074112                          EFT1

148
149                             ;ERROR  61      ERRONEOUS IVC ERROR DURING SEEK COMMAND -
150                             :               VOLUME VALID IS STIL SET
151
002372  070402                          EMT61
002374  073676                          EHT1
002376  074022                          EDT1
002400  074112                          EFT1

152
153                             ;ERROR  62      MOL IS ZERO, BUT OPI WAS NOT
154                             :               REPORTED DURING SEEK COMMAND
155
002402  070422                          EMT62
002404  073676                          EHT1
002406  074022                          EDT1
002410  074112                          EFT1

156
157                             ;ERROR  63      UNUSED
158
002412  000000                          0
002414  000000                          0
002416  000000                          0
002420  000000                          0

159
160                             ;ERROR  64      DRIVE DID NOT DETECT "IVC" ERROR DURING SEEK
161
002422  070440                          EMT64
002424  073676                          EHT1
002426  074022                          EDT1
002430  074112                          EFT1

162
163                             ;ERROR  65      DRIVE EXECUTED A SEEK WITH ERROR SET
164
002432  070460                          EMT65
002434  073676                          EHT1
002436  074022                          EDT1
002440  074112                          EFT1

165
166                             ;ERROR  66      UNEXPECTED ERROR SET IN RMER1
167
002442  070500                          EMT66
002444  073676                          EHT1
002446  074022                          EDT1
002450  074112                          EFT1

168
169                             ;ERROR  67      UNEXPECTED ERROR SET IN RMER2
170
002452  070512                          EMT67
```

```
002454  073676                          EHT1
002456  074022                          EDT1
002460  074112                          EFT1

171
172                             ;ERROR  70      ERRONEOUS "IAE" ERROR DURING RECALIBRATE
173
002462  070524                          EMT70
002464  073676                          EHT1
002466  074022                          EDT1
002470  074112                          EFT1

174
175                             ;ERROR  71      "ILF" ERROR DURING RECALIBRATE
176
002472  070534                          EMT71
002474  073676                          EHT1
002476  074022                          EDT1
002500  074112                          EFT1

177
178                             ;ERROR  72      "OPI" ERROR DURING RECALIBRATE DUE TO "MOL" = 0
179
002502  070544                          EMT72
002504  073676                          EHT1
002506  074022                          EDT1
002510  074112                          EFT1

180
181                             ;ERROR  73      "OPI" ERROR DURING RECALIBRATE BECAUSE ON
182                             ;               CYLINDER DIDN'T DROP
183
002512  070562                          EMT73
002514  073676                          EHT1
002516  074022                          EDT1
002520  074112                          EFT1

184
185                             ;ERROR  74      "IVC" ERROR DURING RECALIBRATE - "VV" - 0
186
002522  070600                          EMT74
002524  073676                          EHT1
002526  074022                          EDT1
002530  074112                          EFT1

187
188                             ;ERROR  75      ERRONEOUS "IVC" ERROR DURING RECALIBRATE - "VV" - 1
189
002532  070610                          EMT75
002534  073676                          EHT1
002536  074022                          EDT1
002540  074112                          EFT1

190
191                             ;ERROR  76      "SKI" ERROR DURING RECALIBRATE
192
002542  070630                          EMT76
```

```
002544  073676                          EHT1
002546  074022                          EDT1
002550  074112                          EFT1

193
194                             ;ERROR  77      "DVC" OCCURRED DURING RECALIBRATE
195
002552  070640                          EMT77
002554  073676                          EHT1
002556  074022                          EDT1
002560  074112                          EFT1

196
197                             ;ERROR  100     LOST "MOL" DURING RECALIBRATE - "OPI" = 0
198
002562  070652                          EMT100
002564  073676                          EHT1
002566  074022                          EDT1
002570  074112                          EFT1

199
200                             ;ERROR  101     LOST "VV" DURING RECALIBRATE - "IVC" = 0
201
002572  070670                          EMT101
002574  073676                          EHT1
002576  074022                          EDT1
002600  074112                          EFT1

202
203                             ;ERROR  102     "ATA" DID NOT SET DURING RECALIBRATE
204
002602  070706                          EMT102
002604  073676                          EHT1
002606  074022                          EDT1
002610  074112                          EFT1

205
206                             ;ERROR  103     "OM" DID NOT RESET DURING RECALIBRATE
207
002612  070716                          EMT103
002614  073676                          EHT1
002616  074022                          EDT1
002620  074112                          EFT1

208
209                             ;ERROR  104     "PIP" IS STIL SET AFTER RECALIBRATE
210
002622  070730                          EMT104
002624  073676                          EHT1
002626  074022                          EDT1
002630  074112                          EFT1

211
212                             ;ERROR  105     UNEXPECTED "ILR" ERROR DURING RECALIBRATE
213
002632  070746                          EMT105
002634  073676                          EHT1
```

```
002636  074022                          EDT1
002640  074112                          EFT1

214
215                             ;ERROR  106     UNEXPECTED ''RMR'' ERROR DURING RECALIBRATE
216
002642  070756                          EMT106
002644  073676                          EHT1
002646  074022                          EDT1
002650  074112                          EFT1

217
218                             ;ERROR  107     ''UNS'' ERROR DURING RECALIBRATE - AC POWER IS LOW
219
002652  070766                          EMT107
002654  073676                          EHT1
002656  074022                          EDT1
002660  074112                          EFT1

220
221                             ;ERROR  110     CANNOT ACCESS MASSBUS CONTROLLER VIA UNIBUS
222
002662  071006                          EMT110
002664  073720                          EHT110
002666  074040                          EDT110
002670  074130                          EFT110

223
224                             ;ERROR  111     NONEXISTENT DEVICE
225
002672  071020                          EMT111
002674  073724                          EHT111
002676  074042                          EDT111
002700  074132                          EFT111

226
227                             ;ERROR  112     DEVICE NOT AVAILABLE
228
002702  071026                          EMT112
002704  073724                          EHT111
002706  074042                          EDT111
002710  074132                          EFT111

229
230                             ;ERROR  113     BUS TIMEOUT-NED STATUS FAILURE
231
002712  071034                          EMT113
002714  000000                          0
002716  000000                          0
002720  000000                          0

232
233                             ;ERROR  114     DEVICE NOT AN RM05/3/2
234
002722  071050                          EMT114
002724  073730                          EHT114
002726  074044                          EDT114
```

```
        002730  074134                          EFT114

    235
    236                                 ;ERROR  115     RMCS1 NOT INITIALIZED BY UNIBUS
    237
        002732  071056                          EMT115
        002734  073676                          EHT1
        002736  074022                          EDT1
        002740  074112                          EFT1

    238
    239                                 ;ERROR  116     RMBA NOT INITIALIZED BY UNIBUS
    240
        002742  071066                          EMT116
        002744  073676                          EHT1
        002746  074022                          EDT1
        002750  074112                          EFT1

    241
    242                                 ;ERROR  117     RMCS2 NOT INITIALIZED BY UNIBUS
    243
        002752  071076                          EMT117
        002754  073676                          EHT1
        002756  074022                          EDT1
        002760  074112                          EFT1

    244
    245                                 ;ERROR  120     RMER1 NOT INITIALIZED BY UNIBUS
    246
        002762  071106                          EMT120
        002764  073676                          EHT1
        002766  074022                          EDT1
        002770  074112                          EFT1

    247
    248                                 ;ERROR  121     RMAS NOT INITIALIZED BY UNIBUS
    249
        002772  071116                          EMT121
        002774  073676                          EHT1
        002776  074022                          EDT1
        003000  074112                          EFT1

    250
    251                                 ;ERROR  122     RMMR1 NOT INITIALIZED BY UNIBUS
    252
        003002  071126                          EMT122
        003004  073676                          EHT1
        003006  074022                          EDT1
        003010  074112                          EFT1

    253
    254                                 ;ERROR  123     RMDS NOT INITIALIZED BY UNIBUS
    255
        003012  071136                          EMT123
        003014  073676                          EHT1
        003016  074022                          EDT1
        003020  074112                          EFT1
```

```
256
257                                     ;ERROR  124     RMEC2 NOT INITIALIZED BY UNIBUS
258
    003022  071146                              EMT124
    003024  073676                              EHT1
    003026  074022                              EDT1
    003030  074112                              EFT1

259
260                                     ;ERROR  125     RMMR2 NOT INITIALIZED BY UNIBUS
261
    003032  071156                              EMT125
    003034  073676                              EHT1
    003036  074022                              EDT1
    003040  074112                              EFT1

262
263                                     ;ERROR  126     RMCS1 NOT CLEARED BY CONTROLLER CLEAR
264
    003042  071166                              EMT126
    003044  073676                              EHT1
    003046  074022                              EDT1
    003050  074112                              EFT1

265
266                                     ;ERROR  127     RMBA NOT CLEARED BY CONTROLLER CLEAR
267
    003052  071200                              EMT127
    003054  073676                              EHT1
    003056  074022                              EDT1
    003060  074112                              EFT1

268
269                                     ;ERROR  130     RMCS2 NOT CLEARED BY CONTROLLER CLEAR
270
    003062  071212                              EMT130
    003064  073676                              EHT1
    003066  074022                              EDT1
    003070  074112                              EFT1

271
272                                     ;ERROR  131     RMER1 NOT CLEARED BY CONTROLLER CLEAR
273
    003072  071224                              EMT131
    003074  073676                              EHT1
    003076  074022                              EDT1
    003100  074112                              EFT1

274
275                                     ;ERROR  132     RMAS NOT CLEARED BY CONTROLLER CLEAR
276
    003102  071236                              EMT132
    003104  073676                              EHT1
    003106  074022                              EDT1
    003110  074112                              EFT1
```

```
277
278                                 ;ERROR  133     RMMR1 NOT CLEARED BY CONTROLLER CLEAR
279
    003112  071250                          EMT133
    003114  073676                          EHT1
    003116  074022                          EDT1
    003120  074112                          EFT1

280
281                                 ;ERROR  134     RMDS NOT CLEARED BY CONTROLLER CLEAR
282
    003122  071262                          EMT134
    003124  073676                          EHT1
    003126  074022                          EDT1
    003130  074112                          EFT1

283
284                                 ;ERROR  135     RMEC2 NOT CLEARED BY CONTROLLER CLEAR
285
    003132  071274                          EMT135
    003134  073676                          EHT1
    003136  074022                          EDT1
    003140  074112                          EFT1

286
287                                 ;ERROR  136     RMMR2 NOT CLEARED BY CONTROLLER CLEAR
288
    003142  071306                          EMT136
    003144  073676                          EHT1
    003146  074022                          EDT1
    003150  074112                          EFT1

289
290                                 ;ERROR  137     RMCS1 NOT CLEARED BY ERROR CLEAR
291
    003152  071320                          EMT137
    003154  073676                          EHT1
    003156  074022                          EDT1
    003160  074112                          EFT1

292
293                                 ;ERROR  140     RMCS2 NOT CLEARED BY ERROR CLEAR
294
    003162  071330                          EMT140
    003164  073676                          EHT1
    003166  074022                          EDT1
    003170  074112                          EFT1

295
296                                 ;ERROR  141     RMCS1 NOT CLEARED BY DRIVE CLEAR
297
    003172  071340                          EMT141
    003174  073676                          EHT1
    003176  074022                          EDT1
    003200  074112                          EFT1

298
```

```
299                                     ;ERROR  142     RMDS NOT CLEARED BY DRIVE CLEAR
300
    003202  071350                                EMT142
    003204  073676                                EHT1
    003206  074022                                EDT1
    003210  074112                                EFT1

301
302                                     ;ERROR  143     RMER1 NOT CLEARED BY DRIVE CLEAR
303
    003212  071360                                EMT143
    003214  073676                                EHT1
    003216  074022                                EDT1
    003220  074112                                EFT1

304
305                                     ;ERROR  144     RMAS NOT CLEARED BY DRIVE CLEAR
306
    003222  071370                                EMT144
    003224  073676                                EHT1
    003226  074022                                EDT1
    003230  074112                                EFT1

307
308                                     ;ERROR  145     RMMR1  NOT CLEARED BY DRIVE CLEAR
309
    003232  071400                                EMT145
    003234  073676                                EHT1
    003236  074022                                EDT1
    003240  074112                                EFT1

310
311                                     ;ERROR  146     RMMR2 NOT CLEARED BY DRIVE CLEAR
312
    003242  071410                                EMT146
    003244  073676                                EHT1
    003246  074022                                EDT1
    003250  074112                                EFT1

313
314                                     ;ERROR  147     RMER2 NOT CLEARED BY DRIVE CLEAR
315
    003252  071420                                EMT147
    003254  073676                                EHT1
    003256  074022                                EDT1
    003260  074112                                EFT1

316
317                                     ;ERROR  150     RMEC2 NOT CLEARED BY DRIVE CLEAR
318
    003262  071430                                EMT150
    003264  073676                                EHT1
    003266  074022                                EDT1
    003270  074112                                EFT1

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

```
321
    003272  071440                          EMT151
    003274  073676                          EHT1
    003276  074022                          EDT1
    003300  074112                          EFT1

322
323                                 ;ERROR  152     DRIVE FAULT
324
    003302  071452                          EMT152
    003304  073676                          EHT1
    003306  074022                          EDT1
    003310  074112                          EFT1

325
326                                 ;ERROR  153     UNSAFE SHOULD BE SET BECAUSE DVC IS SET
327
    003312  071464                          EMT153
    003314  073676                          EHT1
    003316  074022                          EDT1
    003320  074112                          EFT1

328
329                                 ;ERROR  154     UNSAFE SHOULD NOT BE SET, AC IS LOW
330
    003322  071502                          EMT154
    003324  073676                          EHT1
    003326  074022                          EDT1
    003330  074112                          EFT1

331
332                                 ;ERROR  155     VOLUME VALID NOT SET BY PACK ACK
333
    003332  071520                          EMT155
    003334  073676                          EHT1
    003336  074022                          EDT1
    003340  074112                          EFT1

334
335                                 ;ERROR  156     OFFSET MODE NOT SET BY OFFSET COMMAND
336
    003342  071532                          EMT156
    003344  073676                          EHT1
    003346  074022                          EDT1
    003350  074112                          EFT1

337
338                                 ;ERROR  157     OFFSET MODE NOT RESET BY RTC COMMAND
339
    003352  071544                          EMT157
    003354  073676                          EHT1
    003356  074022                          EDT1
    003360  074112                          EFT1

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

```
003362  071556                          EMT160
003364  073676                          EHT1
003366  074022                          EDT1
003370  074112                          EFT1

343
344                             ;ERROR  161     RMDA NOT RESET BY RIP COMMAND
345
003372  071566                          EMT161
003374  073676                          EHT1
003376  074022                          EDT1
003400  074112                          EFT1

346
347                             ;ERROR  162     RMDC NOT RESET BY RIP COMMAND
348
003402  071600                          EMT162
003404  073676                          EHT1
003406  074022                          EDT1
003410  074112                          EFT1

349
350                             ;ERROR 163      DATA WAS ECC CORRECTED BUT DOES NOT COMPARE WITH
351                             ;               WRITE BUFFER
352
003412  073472                          EMT336
003414  073760                          EHT336
003416  074056                          EDT336
003420  074146                          EFT336

353
354                             ;ERROR  164     OPI SHOULD NOT BE SET
355
003422  071622                          EMT164
003424  073676                          EHT1
003426  074022                          EDT1
003430  074112                          EFT1

356
357                             ;ERROR  165     IVC SHOULD NOT BE SET
358
003432  071630                          EMT165
003434  073676                          EHT1
003436  074022                          EDT1
003440  074112                          EFT1

359
360                             ;ERROR  166     IAE SHOULD NOT BE SET
361
003442  071636                          EMT166
003444  073676                          EHT1
003446  074022                          EDT1
003450  074112                          EFT1

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

```
003452  071644                          EMT167
003454  073676                          EHT1
003456  074022                          EDT1
003460  074112                          EFT1

365
366                             ;ERROR 170      UNUSED
367
003462  000000                          0
003464  000000                          0
003466  000000                          0
003470  000000                          0

368
369                             ;ERROR  171     "ATA" NOT SET DURING RETURN TO CENTERLINE
370
003472  071662                          EMT171
003474  073676                          EHT1
003476  074022                          EDT1
003500  074112                          EFT1

371
372                             ;ERROR  172     "ATA" NOT SET BY OFFSET COMMAND
373
003502  071672                          EMT172
003504  073676                          EHT1
003506  074022                          EDT1
003510  074112                          EFT1

374
375                             ;ERROR  173     RMER2 NOT INITIALIZED BY UNIBUS INIT
376
003512  071702                          EMT173
003514  073676                          EHT1
003516  074022                          EDT1
003520  074112                          EFT1

377
378                             ;ERROR  174     RMER2 NOT INITIALIZED BY CONTROLLER CLEAR
379
003522  071712                          EMT174
003524  073676                          EHT1
003526  074022                          EDT1
003530  074112                          EFT1

380
381                             ;ERROR  175     SELECTED DEVICE IS IN WRITE PROTECT
382
003532  071724                          EMT175
003534  073676                          EHT1
003536  074022                          EDT1
003540  074112                          EFT1

383
384                             ;ERROR  176     CANNOT SET DIAGNOSTIC MODE
385
003542  071732                          EMT176
```

```
003544  073676                          EHT1
003546  074022                          EDT1
003550  074112                          EFT1

386
387                             ;ERROR  177     INCORRECT "MOL" STATUS DURING DIAGNOSTIC MODE
388
003552  071740                          EMT177
003554  073676                          EHT1
003556  074022                          EDT1
003560  074112                          EFT1

389
390                             ;ERROR  200     INCORRECT "PIP" STATUS DURING DIAGNOSTIC MODE
391
003562  071752                          EMT200
003564  073676                          EHT1
003566  074022                          EDT1
003570  074112                          EFT1

392
393                             ;ERROR  201     INCORRECT "WRL" STATUS DURING DIAGNOSTIC MODE
394
003572  071764                          EMT201
003574  073676                          EHT1
003576  074022                          EDT1
003600  074112                          EFT1

395
396                             ;ERROR  202     INCORRECT "SKI" STATUS DURING DIAGNOSTIC MODE
397
003602  071776                          EMT202
003604  073676                          EHT1
003606  074022                          EDT1
003610  074112                          EFT1

398
399                             ;ERROR  203     INCORRECT "DVC" STATUS DURING DIAGNOSTIC MODE
400
003612  072010                          EMT203
003614  073676                          EHT1
003616  074022                          EDT1
003620  074112                          EFT1

401
402                             ;ERROR  204     "VV" WAS NOT RESET BY MAINTENANCE UNIT READY
403
003622  072022                          EMT204
003624  073676                          EHT1
003626  074022                          EDT1
003630  074112                          EFT1

404
405                             ;ERROR  205     SELECTED DEVICE HAS A PERSISTENT "SKI" ERROR
406
003632  072040                          EMT205
003634  073676                          EHT1
```

```
003636  074022                          EDT1
003640  074112                          EFT1

407
408                             ;ERROR  206     "LBC" DID NOT SET DURING DIAGNOSTIC MODE
409
003642  072050                          EMT206
003644  073676                          EHT1
003646  074022                          EDT1
003650  074112                          EFT1

410
411                             ;ERROR  207     UNEXPECTED LOSS OF "MOL" - MEDIUM IS OFF LINE
412
003652  072060                          EMT207
003654  073676                          EHT1
003656  074022                          EDT1
003660  074112                          EFT1

413
414                             ;ERROR  210     UNEXPECTED LOSS OF VOLUME VALID - "VV" = 0
415
003662  072072                          EMT210
003664  073676                          EHT1
003666  074022                          EDT1
003670  074112                          EFT1

416
417                             ;ERROR  211     UNEXPECTED MECHANICAL MOTION - "PIP" - 1
418
003672  072100                          EMT211
003674  073676                          EHT1
003676  074022                          EDT1
003700  074112                          EFT1

419
420                             ;ERROR  212     UNEXPECTED DEVICE FAULT - "DVC" = 1
421
003702  072114                          EMT212
003704  073676                          EHT1
003706  074022                          EDT1
003710  074112                          EFT1

422
423                             ;ERROR  213     UNEXPECTED SEEK INCOMPLETE ERROR - "SKI" = 1
424
003712  072130                          EMT213
003714  073676                          EHT1
003716  074022                          EDT1
003720  074112                          EFT1

425
426                             ;ERROR  214     DRIVE EXECUTED A RECALIBRATE WITH ERROR SET
427
003722  072140                          EMT214
003724  073710                          EHT2
003726  074032                          EDT2
```

```
        003730  074122                          EFT2

    428
    429                                 ;ERROR  215     DRIVE DID NOT DETECT "IVC" ERROR DURING RECALIBRATE
    430
        003732  072160                          EMT215
        003734  073710                          EHT2
        003736  074032                          EDT2
        003740  074122                          EFT2

    431
    432                                 ;ERROR  216     INCORRECT "IVC" STATUS
    433
        003742  072172                          EMT216
        003744  073676                          EHT1
        003746  074022                          EDT1
        003750  074112                          EFT1

    434
    435                                 ;ERROR  217     INCORRECT "IAE" STATUS
    436
        003752  072202                          EMT217
        003754  073676                          EHT1
        003756  074022                          EDT1
        003760  074112                          EFT1

    437
    438                                 ;ERROR 220      INCORRECT "WLE" STATUS
    439
        003762  072212                          EMT220
        003764  073676                          EHT1
        003766  074022                          EDT1
        003770  074112                          EFT1

    440
    441                                 ;ERROR  221     INCORRECT "OPI" STATUS
    442
        003772  072222                          EMT221
        003774  073676                          EHT1
        003776  074022                          EDT1
        004000  074112                          EFT1

    443
    444                                 ;ERROR  222     RM DID NOT DETECT RMR ERROR
    445
        004002  072232                          EMT222
        004004  073676                          EHT1
        004006  074022                          EDT1
        004010  074112                          EFT1

    446
    447                                 ;ERROR  223     RM DID NOT DETECT PARITY ERROR ON MASSBUS CONTROL BUS
    448
        004012  072242                          EMT223
        004014  073734                          EHT223
        004016  074046                          EDT223
        004020  074136                          EFT223
```

```
449
450                                     ;ERROR  224     UNUSED
451
    004022  000000                              0
    004024  000000                              0
    004026  000000                              0
    004030  000000                              0

452
453                                     ;ERROR  225     UNUSED
454
    004032  000000                              0
    004034  000000                              0
    004036  000000                              0
    004040  000000                              0

455
456                                     ;ERROR  226     UNUSED
457
    004042  000000                              0
    004044  000000                              0
    004046  000000                              0
    004050  000000                              0

458
459                                     ;ERROR  227     UNUSED
460
    004052  000000                              0
    004054  000000                              0
    004056  000000                              0
    004060  000000                              0

461
462                                     ;ERROR  230     UNUSED
463
    004062  000000                              0
    004064  000000                              0
    004066  000000                              0
    004070  000000                              0

464
465                                     ;ERROR  231     UNUSED
466
    004072  000000                              0
    004074  000000                              0
    004076  000000                              0
    004100  000000                              0

467
468                                     ;ERROR  232     UNUSED
469
    004102  000000                              0
    004104  000000                              0
    004106  000000                              0
    004110  000000                              0
```

```
470
471                                     ;ERROR  233     UNUSED
472
    004112  000000                              0
    004114  000000                              0
    004116  000000                              0
    004120  000000                              0

473
474                                     ;ERROR  234     UNUSED
475
    004122  000000                              0
    004124  000000                              0
    004126  000000                              0
    004130  000000                              0

476
477                                     ;ERROR  235     UNUSED
478
    004132  000000                              0
    004134  000000                              0
    004136  000000                              0
    004140  000000                              0

479
480                                     ;ERROR  236     UNUSED
481
    004142  000000                              0
    004144  000000                              0
    004146  000000                              0
    004150  000000                              0

482
483                                     ;ERROR  237     UNUSED
484
    004152  000000                              0
    004154  000000                              0
    004156  000000                              0
    004160  000000                              0

485
486                                     ;ERROR  240     UNUSED
487
    004162  000000                              0
    004164  000000                              0
    004166  000000                              0
    004170  000000                              0

488
489                                     ;ERROR  241     UNUSED
490
    004172  000000                              0
    004174  000000                              0
    004176  000000                              0
    004200  000000                              0

491
```

```
                                        ;ERROR  242     UNUSED
                                                0
004202  000000                                  0
004204  000000                                  0
004206  000000                                  0
004210  000000                                  0

                                        ;ERROR  243     UNUSED
                                                0
004212  000000                                  0
004214  000000                                  0
004216  000000                                  0
004220  000000                                  0

                                        ;ERROR  244     UNUSED
                                                0
004222  000000                                  0
004224  000000                                  0
004226  000000                                  0
004230  000000                                  0

                                        ;ERROR  245     UNUSED
                                                0
004232  000000                                  0
004234  000000                                  0
004236  000000                                  0
004240  000000                                  0

                                        ;ERROR  246     "ATA" NOT RESET BY GO WHEN "ERR" - 0
004242  072316                                  EMT246
004244  073676                                  EHT1
004246  074022                                  EDT1
004250  074112                                  EFT1

                                        ;ERROR  247     "ATA" NOT RESET BY WRITING RMAS
004252  072326                                  EMT247
004254  073676                                  EHT1
004256  074022                                  EDT1
004260  074112                                  EFT1

                                        ;ERROR  250     "ATA" WAS RESET BY GO WHEN "ERR" - 1
004262  072340                                  EMT250
004264  073676                                  EHT1
004266  074022                                  EDT1
004270  074112                                  EFT1

                                        ;ERROR  251     PROGRAM INTERRUPT WAS NOT GENERATED
```

```
514
    004272  072354                          EMT251
    004274  073710                          EHT2
    004276  074032                          EDT2
    004300  074122                          EFT2

515
516                             ;ERROR  252     PROGRAM INTERRUPT SHOULD NOT HAVE BEEN GENERATED
517
    004302  072362                          EMT252
    004304  073710                          EHT2
    004306  074032                          EDT2
    004310  074122                          EFT2

518
519                             ;ERROR  253     OFFSET MODE WAS NOT RESET BY WRITING RMDC
520
    004312  072370                          EMT253
    004314  073676                          EHT1
    004316  074022                          EDT1
    004320  074112                          EFT1

521
522                             ;ERROR  254     INCORRECT "ILF" STATUS
523
    004322  072406                          EMT254
    004324  073676                          EHT1
    004326  074022                          EDT1
    004330  074112                          EFT1

524
525                             ;ERROR  255     INCORRECT "ATA" STATUS
526
    004332  072416                          EMT255
    004334  073676                          EHT1
    004336  074022                          EDT1
    004340  074112                          EFT1

527
528                             ;ERROR  256     INCORRECT "ILR" STATUS
529
    004342  072426                          EMT256
    004344  073746                          EHT256
    004346  074046                          EDT223
    004350  074136                          EFT223

530
531                             ;ERROR  257     INVALID IAE STATUS DURING SEARCH COMMAND
532
    004352  072436                          EMT257
    004354  073676                          EHT1
    004356  074022                          EDT1
    004360  074112                          EFT1

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

```
004362  072450                          EMT260
004364  073676                          EHT1
004366  074022                          EDT1
004370  074112                          EFT1

536
537                             ;ERROR  261     DRIVE EXECUTED SEARCH WITH ERROR SET
538
004372  072462                          EMT261
004374  073676                          EHT1
004376  074022                          EDT1
004400  074112                          EFT1

539
540                             ;ERROR  262     "LBC" ERROR NOT SET DURING DIAGNOSTIC MODE
541
004402  072502                          EMT262
004404  073676                          EHT1
004406  074022                          EDT1
004410  074112                          EFT1

542
543                             ;ERROR  263     "SKI" ERROR DURING SEARCH COMMAND
544
004412  072512                          EMT263
004414  073676                          EHT1
004416  074022                          EDT1
004420  074112                          EFT1

545
546                             ;ERROR  264     "IVC" ERROR DURING SEARCH - LOST VOLUME VALID
547
004422  072522                          EMT264
004424  073676                          EHT1
004426  074022                          EDT1
004430  074112                          EFT1

548
549                             ;ERROR  265     ERRONEOUS IVC ERROR DURING SEARCH - VOLUME IS VALID
550
004432  072542                          EMT265
004434  073676                          EHT1
004436  074022                          EDT1
004440  074112                          EFT1

551
552                             ;ERROR  266     DEVICE FAULT (DVC) DURING SEARCH
553
004442  072562                          EMT266
004444  073676                          EHT1
004446  074022                          EDT1
004450  074112                          EFT1

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

```
004452  072574                          EMT267
004454  073676                          EHT1
004456  074022                          EDT1
004460  074112                          EFT1

558
559                             ;ERROR  270     OPI ERROR DURING SEARCH BECAUSE MOL - 0
560
004462  072612                          EMT270
004464  073676                          EHT1
004466  074022                          EDT1
004470  074112                          EFT1

561
562                             ;ERROR  271     OPI ERROR DURING SEARCH BECAUSE ON CYLINDER
563                             ;               DIDN'T DROP
564
004472  072626                          EMT271
004474  073676                          EHT1
004476  074022                          EDT1
004500  074112                          EFT1

565
566                             ;ERROR  272     LOST MOL DURING SEARCH, OPI IS NOT SET
567
004502  072644                          EMT272
004504  073676                          EHT1
004506  074022                          EDT1
004510  074112                          EFT1

568
569                             ;ERROR  273     PIP STIL SET AFTER SEARCH
570
004512  072662                          EMT273
004514  073676                          EHT1
004516  074022                          EDT1
004520  074112                          EFT1

571
572                             ;ERROR  274     PARITY ERROR OCCURRED WHILE WRITING REMOTE
573                             ;               REGISTERS BUT MXF DID NOT SET
574
004522  072700                          EMT274
004524  073676                          EHT1
004526  074022                          EDT1
004530  074112                          EFT1

575
576                             ;ERROR  275     MXF ERROR - COMPOSITE ERROR OCCURRED BEFORE DATA
577                             ;               COMMAND STARTED
578
004532  072716                          EMT275
004534  073676                          EHT1
004536  074022                          EDT1
004540  074112                          EFT1

579
```

```
                                ;ERROR  276     "OPI" ERROR DURING DATA TRANSFER BECAUSE "MOL" WAS
                                ;               ZERO
004542  072730                          EMT276
004544  073676                          EHT1
004546  074022                          EDT1
004550  074112                          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
004552  072744                          EMT277
004554  073676                          EHT1
004556  074022                          EDT1
004560  074112                          EFT1

                                ;ERROR  300     "IVC" ERROR DURING DATA TRANSFR BECAUSE VOLUME
                                ;               WAS NOT VALID
004562  072762                          EMT300
004564  073676                          EHT1
004566  074022                          EDT1
004570  074112                          EFT1

                                ;ERROR  301     ERRONEOUS "IVC" ERROR DURING DATA TRANSFER - VOLUME
                                ;               IS VALID
004572  073002                          EMT301
004574  073676                          EHT1
004576  074022                          EDT1
004600  074112                          EFT1

                                ;ERROR  302     "ILR" ERROR DURING DATA TRANSFER
004602  073024                          EMT302
004604  073676                          EHT1
004606  074022                          EDT1
004610  074112                          EFT1

                                ;ERROR  303     "ILF" ERROR DURING DATA TRANSFER
004612  073036                          EMT303
004614  073676                          EHT1
004616  074022                          EDT1
004620  074112                          EFT1

                                ;ERROR  304     "RMR" ERROR DURING DATA TRANSFER
004622  073050                          EMT304
004624  073676                          EHT1
```

```
004626  074022                          EDT1
004630  074112                          EFT1

605
606                             ;ERROR  305     INCORRECT "IAE" STATUS DURING DATA TRANSFER
607
004632  073062                          EMT305
004634  073676                          EHT1
004636  074022                          EDT1
004640  074112                          EFT1

608
609                             ;ERROR  306     "SKI" ERROR DURING DATA TRANSFER
610
004642  073074                          EMT306
004644  073676                          EHT1
004646  074022                          EDT1
004650  074112                          EFT1

611
612                             ;ERROR  307     DRIVE DID NOT DETECT SKI ERROR DUE TO CYLINDER
613
004652  073104                          EMT307
004654  073676                          EHT1
004656  074022                          EDT1
004660  074112                          EFT1

614
615                             ;ERROR  310     DEVICE FAULT DURING DATA TRANSFER
616
004662  073124                          EMT310
004664  073676                          EHT1
004666  074022                          EDT1
004670  074112                          EFT1

617
618                             ;ERROR  311     LOSS OF BIT CLOCK DURING DATA TRANSFER
619
004672  073136                          EMT311
004674  073676                          EHT1
004676  074022                          EDT1
004700  074112                          EFT1

620
621                             ;ERROR  312     LOSS OF SYSTEM CLOCK DURING DATA TRANSFER
622
004702  073150                          EMT312
004704  073676                          EHT1
004706  074022                          EDT1
004710  074112                          EFT1

623
624                             ;ERROR  313     UNSAFE ERROR DURING DATA TRANSFER (DVC   0)
625
004712  073162                          EMT313
004714  073676                          EHT1
004716  074022                          EDT1
```

```
004720  074112                          EFT1

626
627                             ;ERROR  314     DRIVE TIMING ERROR DURING DATA TRANSFER
628
004722  073202                          EMT314
004724  073676                          EHT1
004726  074022                          EDT1
004730  074112                          EFT1

629
630                             ;ERROR  315     WRITE LOCK ERROR
631
004732  073214                          EMT315
004734  073676                          EHT1
004736  074022                          EDT1
004740  074112                          EFT1

632
633                             ;ERROR  316     ERRONEOUS WRITE LOCK ERROR
634
004742  073226                          EMT316
004744  073676                          EHT1
004746  074022                          EDT1
004750  074112                          EFT1

635
636                             ;ERROR  317     HEADER CRC ERROR DURING DATA TRANSFER
637
004752  073240                          EMT317
004754  073676                          EHT1
004756  074022                          EDT1
004760  074112                          EFT1

638
639                             ;ERROR  320     FORMAT ERROR DURING DATA TRANSFER
640
004762  073250                          EMT320
004764  073676                          EHT1
004766  074022                          EDT1
004770  074112                          EFT1

641
642                             ;ERROR  321     HEADER COMPARE ERROR DURING DATA TRANSFER
643
004772  073260                          EMT321
004774  073676                          EHT1
004776  074022                          EDT1
005000  074112                          EFT1

644
645                             ;ERROR  322     HEADER ERRORS SHOULD NOT BE SET
646
005002  073270                          EMT322
005004  073676                          EHT1
005006  074022                          EDT1
005010  074112                          EFT1
```

```
647
648                                     ;ERROR  323     DATA CHECK ERROR DURING DATA TRANSFER
649
    005012  073276                              EMT323
    005014  073676                              EHT1
    005016  074022                              EDT1
    005020  074112                              EFT1

650
651                                     ;ERROR  324     CORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
652
    005022  073306                              EMT324
    005024  073676                              EHT1
    005026  074022                              EDT1
    005030  074112                              EFT1

653
654                                     ;ERROR  325     UNCORRECTABLE DATA CHECK ERROR DURING DATA TRANSFER
655
    005032  073320                              EMT325
    005034  073676                              EHT1
    005036  074022                              EDT1
    005040  074112                              EFT1

656
657                                     ;ERROR  326     DATA PARITY ERROR DURING READ COMMMAND
658
    005042  073332                              EMT326
    005044  073676                              EHT1
    005046  074022                              EDT1
    005050  074112                              EFT1

659
660                                     ;ERROR  327     OFFSET MODE NOT RESET BY WRITE COMMAND
661
    005052  073350                              EMT327
    005054  073676                              EHT1
    005056  074022                              EDT1
    005060  074112                              EFT1

662
663                                     ;ERROR  330     DATA PARITY ERROR DURING WRITE COMMAND
664
    005062  073362                              EMT330
    005064  073676                              EHT1
    005066  074022                              EDT1
    005070  074112                              EFT1

665
666                                     ;ERROR  331     WRITE CLOCK FALURE DURING WRITE COMMAND
667
    005072  073372                              EMT331
    005074  073676                              EHT1
    005076  074022                              EDT1
    005100  074112                              EFT1
```

```
668
669                                     ;ERROR  332     DATA LATE ERROR DURING DATA TRANSFER
670
    005102  073404                                    EMT332
    005104  073676                                    EHT1
    005106  074022                                    EDT1
    005110  074112                                    EFT1

671
672                                     ;ERROR  333     PIP STIL SET AFTER DATA TRANSFER - SKI = 0
673
    005112  073416                                    EMT333
    005114  073676                                    EHT1
    005116  074022                                    EDT1
    005120  074112                                    EFT1

674
675                                     ;ERROR  334     LOST MOL DURING DATA TRANSFER - OPI = 0
676
    005122  073434                                    EMT334
    005124  073676                                    EHT1
    005126  074022                                    EDT1
    005130  074112                                    EFT1

677
678                                     ;ERROR  335     LOST VOLUME VALID DURING DATA TRANSFER - IVC  = 0
679
    005132  073452                                    EMT335
    005134  073676                                    EHT1
    005136  074022                                    EDT1
    005140  074112                                    EFT1

680
681                                     ;ERROR  336     DATA READ DOES NOT COMPARE WITH DATA WRITTEN
682
    005142  073472                                    EMT336
    005144  073760                                    EHT336
    005146  074056                                    EDT336
    005150  074146                                    EFT336

683
684                                     ;ERROR  337     WRITE CHECK ERROR NOT DETECTED
685
    005152  073502                                    EMT337
    005154  073772                                    EHT337
    005156  074066                                    EDT337
    005160  074156                                    EFT337

686
687                                     ;ERROR  340     WRITE CHECK ERROR AT UNEXPECTED ADDRESS
688
    005162  073512                                    EMT340
    005164  073760                                    EHT336
    005166  074056                                    EDT336
    005170  074146                                    EFT336

689
```

```
                                        ;ERROR  341     INCORRECT DATA DURING WRITE CHECK ERROR
005172  073524                                  EMT341
005174  073760                                  EHT336
005176  074056                                  EDT336
005200  074146                                  EFT336

                                        ;ERROR  342     "IVC" ERROR NOT DETECTED DURING DATA TRANSFER
005202  073532                                  EMT342
005204  073676                                  EHT1
005206  074022                                  EDT1
005210  074112                                  EFT1

                                        ;ERROR  343     "FER" NOT DETECTED DURING DATA TRANSFER
005212  073544                                  EMT343
005214  073676                                  EHT1
005216  074022                                  EDT1
005220  074112                                  EFT1

                                        ;ERROR  344     "HCE" NOT DETECTED DURING DATA TRANSFER
005222  073556                                  EMT344
005224  074004                                  EHT344
005226  074076                                  EDT344
005230  074166                                  EFT344

                                        ;ERROR  345     "BSE" NOT DETECTED DURING DATA TRANSFER
005232  073570                                  EMT345
005234  073676                                  EHT1
005236  074022                                  EDT1
005240  074112                                  EFT1

                                        ;ERROR  346     HEADER ERROR WAS DETECTED W/ HCI SET
005242  073600                                  EMT346
005244  073676                                  EHT1
005246  074022                                  EDT1
005250  074112                                  EFT1

                                        ;ERROR  347     DATA TRANSFER NOT ABORTED W/ COMP ERROR SET
005252  073614                                  EMT347
005254  073676                                  EHT1
005256  074022                                  EDT1
005260  074112                                  EFT1

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

```
712
    005262  073626                          EMT350
    005264  073676                          EHT1
    005266  074022                          EDT1
    005270  074112                          EFT1

713
714                             ;ERROR  351     "ATA" DID NOT SET DURING SEARCH
715
    005272  073644                          EMT351
    005274  073676                          EHT1
    005276  074022                          EDT1
    005300  074112                          EFT1

716
717                             ;ERROR  352     PROGRAM TIMEOUT WHILE TESTING RMLA
718
    005302  073654                          EMT352
    005304  000000                          0
    005306  000000                          0
    005310  000000                          0

719
720                             ;ERROR  353     LOOK AHEAD TEST FAILS
721
    005312  073660                          EMT353
    005314  074016                          EHT353
    005316  074110                          EDT353
    005320  074176                          EFT353

722
723                             ;ERROR  354     BSE SHOULD NOT BE SET
724
    005322  073670                          EMT354
    005324  073676                          EHT1
    005326  074022                          EDT1
    005330  074112                          EFT1

725
726                             ;PUT ERROR TABLE HERE
727
```

```
                        .SBTTL  ERROR TABLE USAGE

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

```
                                        ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS

005332  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE THE TIME OUT OCCURED
005334  005740                          TST     -(R0)           ;ADJUST PC -2
005336  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
005340  104401  005346                  TYPE    ,65$            ;;TYPE ASCIZ STRING
005344  000417                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
005404                          64$:
005404  010046                          MOV     R0,-(SP)        ;SETUP FOR TYPING OUT PC
005406  104402                          TYPOC
005410  000240                          NOP                     ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
                                                                ;TO STOP ON UNEXPECTED TIMEOUT.


                                .SBTTL  START OF PROGRAM

005412  000240                  START:  NOP
005414  005227  000000                  INC     #0              ;TTY LOOP, WAIT FOR INCREMENT
005420  001375                          BNE     .-4             ;OF WORD
005422  000005                          RESET                   ;RESET THE WORLD


                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
005424  012706  001114                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
005430  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
005432  022706  001154                  CMP     #SWR,R6         ;;DONE?
005436  001374                          BNE     .-6             ;;LOOP BACK IF NO
005440  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
005444  012737  063210  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005452  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005460  012737  063710  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005466  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005474  012737  065450  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005502  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005510  012737  065556  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005516  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005524  013737  042172  042164          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
005532  005037  001206                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005536  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005542  112737  000001  001131          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005550  012737  005550  001122          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005556  012737  005556  001124          MOV     #.,$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.
005564  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005570  012737  005624  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
005576  012737  177570  001154          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005604  012737  177570  001156          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005612  022777  177777  173334          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005620  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005622  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
005624  012716  005632          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
005630  000002                          RTI
005632  012737  000176  001154  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
005640  012737  000174  001156          MOV     #DISPREG,DISPLAY
```

```
        005646  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

        005652  005037  001230                  CLR     $PASS           ;;CLEAR PASS COUNT
        005656  132737  000200  001243          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
        005664  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
        005666  012737  001244  001154          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
        005674                          67$:
    20                                  ;SETUP "TIMEOUT" TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
    21  005674  012737  005332  000004          MOV     #BADTMO,ERRVEC  ;SETUP FOR UNEXPECTED TIMEOUT
    22  005702  012737  000300  000006          MOV     #PR6,ERRVEC+2   ;LEVEL 6
    23  005710  012746  000300                  MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
        005714  012746  005722                  MOV     #68$,-(SP)      ;;PUT NEW PC ON  STACK
        005720  000002                          RTI                     ;;POP NEW PC AND PS
        005722                          68$:
    24
    25                                  .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
        005722  005227  177777                  INC     #-1             ;;FIRST TIME?
        005726  001056                          BNE     69$             ;;BRANCH IF NO
        005730  022737  042326  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
        005736  001452                          BEQ     69$             ;;BRANCH IF YES
        005740  104401  006006                  TYPE    ,70$            ;;TYPE ASCIZ STRING
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
        005744  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
        005750  001012                          BNE     71$             ;;BRANCH IF YES
        005752  123727  001242  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
        005760  001406                          BEQ     71$             ;;BRANCH IF YES
        005762  023727  001154  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
        005770  001005                          BNE     72$             ;;BRANCH IF NO
        005772  104407                          GTSWR                   ;;GET SOFT-SWR SETTINGS
        005774  000403                          BR      72$
        005776  112737  000001  001150  71$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
        006004                          72$:
        006004  000427                          BR      69$             ;;GET OVER THE ASCIZ
                                        ;;70$:  .ASCIZ  <CRLF>@CZRMMA0 - RM05/3/2 FUNCTIONAL TEST, PART 1@<CRLF>
        006064                          69$:
    26
    27                                  ;THE FOLLOWING FINDS OUT THE PROGRAM CONTROL MODE:
    28                                  ;PAPER TAPE (MANUAL), ACT11, XXDP CHAIN OR DUMP
    29
    30  006064  005037  001330                  CLR     XXDP            ;CLEAR 'XXDP' LOAD DEVICE STORAGE
    31  006070  122737  000016  000041          CMPB    #16,@#41        ;LOADED FROM AN RM05/3/2 ?
    32  006076  001160                          BNE     3$              ;BRANCH IF NOT
    33  006100  013737  000040  001330          MOV     @#40,XXDP       ;GET DEVICE INDICATOR AND NUMBER
    34  006106  122737  000007  001330          CMPB    #7,XXDP         ;IS IT A VALID NUMBER ?
    35  006114  103002                          BHIS    1$              ;YES
    36  006116  105037  001330                  CLRB    XXDP            ;NO, DEFAULT TO DRIVE 0
    37  006122  005737  000042          1$:     TST     @#42            ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
    38  006126  001425                          BEQ     2$              ;BR IF NEITHER
    39  006130  104401  006136                  TYPE    ,74$            ;;TYPE ASCIZ STRING
        006134  000412                          BR      73$             ;;GET OVER THE ASCIZ
                                        ;;74$:  .ASCIZ  <CRLF>/NOT TESTING DRIVE /
        006162                          73$:
    40  006162  005046                          CLR     -(SP)           ;CLEAR WORD ON STACK
    41  006164  113716  001330                  MOVB    XXDP,(SP)       ;GET DRIVE ADDRESS
    42  006170  104403                          TYPOS                   ;TYPE THE ADDRESS
    43  006172     001                          .BYTE   1               ;ONLY 1 CHARACTER
```

```
                                        .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    ,76$            ;;TYPE ASCIZ STRING
                                        BR      75$             ;;GET OVER THE ASCIZ
                                ;;76$:  .ASCIZ  <CRLF>/TO TEST DRIVE /
                                75$:
                                        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    ,78$            ;;TYPE ASCIZ STRING
                                        BR      77$             ;;GET OVER THE ASCIZ
                                ;;78$:  .ASCIZ  /, HALT PROGRAM, REMOVE RRDP PACK AND REPLACE IT/<CRLF>
                                77$:
                                        TYPE    ,79$            ;;TYPE ASCIZ STRING
                                        BR      3$              ;;GET OVER THE ASCIZ
                                ;;79$:  .ASCIZ  /WITH A WORK PACK, CLEAR LOCATION 40 AND RESTART PROGRAM./<CRLF>
                                3$:
                                ;CHECK FOR AUTO MODE OR STANDLONE MODE
                                        TSTB    $AUTOB          ;RUNNING IN AUTO MODE ?
                                        BEQ     STANDALONE      ;BR IF NO
                                        MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES

                                ;PROGRAM IS RUNNING IN AUTO MODE - SEE IF SIZING IS ALLOWED
                                XSIZ:
                                        BITB    #BIT7,$ENVM     ;SIZING ALLOWED ?
                                        BNE     7$              ;NO

                                        CLR     R1              ;START FROM DRIVE 0
                                        MOV     $BASE,R0        ;LOAD THE BASE ADDRESS

                                1$:     BITB    ATNTBL(R1),$DEVM        ;IS DEVICE PRESENT IN MAP ?
                                        BEQ     6$              ;BR IF NO
                                        MOV     #LODEV,5$       ;GET ADDRESS OF LOAD DEVICE MESSAGE
                                        TST     XXDP            ;LOADED FROM RM05/3/2 ?
                                        BEQ     2$              ;NO
                                        CMPB    XXDP,R1         ;IS THIS THE DRIVE ?
                                        BEQ     3$              ;YES, TRY NEXT DRIVE

                                2$:     MOV     #CLR,RMCS2(R0)  ;CLEAR MASS BUS
                                        MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
                                        TST     RMDS(R0)        ;TRY TO ACCESS AN RM DRIVE REGISTER
                                        MOV     #NOTPRS,5$      ;GET ADDRESS OF NOT PRESENT MESSAGE
                                        BIT     #NED,RMCS2(R0)  ;IS DRIVE PRESENT ?
                                        BNE     3$              ;BR IF NO
                                        MOV     #NOTAVL,5$      ;GET ADDRESS OF AVAILABLE MESSAGE
                                        BIT     #DVA,RMCS1(R0)  ;IS DRIVE AVAILABLE ?
                                        BEQ     3$              ;BR IF NO
                                        MOV     #OFFLIN,5$      ;GET ADDRESS OF OFF LINE MESSAGE
                                        BIT     #MOL,RMDS(R0)   ;IS MEDIUM ON LINE ?
                                        BEQ     3$              ;BR IF NO
                                        BR      6$
```

```
 95
 96 006616  146137  067146  001300  3$:     BICB    ATNTBL(R1),$DEVM        ;CLEAR DEVICE FROM BIT MAP
 97 006624  104401  001217          4$:     TYPE    ,$CRLF                  ;CR-LF
 98 006630  104401  066737                  TYPE    ,MSGDRV                 ;TYPE 'DRIVE'
 99 006634  010146                          MOV     R1,-(SP)                ;;SAVE R1 FOR TYPEOUT
    006636  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
    006640     002                          .BYTE   2                       ;;TYPE 2 DIGIT(S)
    006641     000                          .BYTE   0                       ;;SUPPRESS LEADING ZEROS
100 006642  104401                          TYPE                            ;TYPE ERROR MESSAGE
101 006644  000000                  5$:     .WORD   0                       ;ADDRESS OF MESSAGE GOES HERE
102
103 006646  005201                  6$:     INC     R1                      ;INCREMENT THE DRIVE ADDRESS
104 006650  020127  000007                  CMP     R1,#7                   ;ALL DRIVES ARE CHECKED ?
105 006654  003706                          BLE     1$                      ;BRANCH IF NOT
106
107 006656  104401  001217          7$:     TYPE    ,$CRLF                  ;CR-LF
108 006662  005004                          CLR     R4                      ;THESE TWO LOOPS ARE ADDED TO
109 006664  005304                          DEC     R4                      ;WAIT FOR TTY TO FINISH TYPING.
110 006666  001376                          BNE     .-2
111 006670  005304                          DEC     R4
112 006672  001376                          BNE     .-2
113
114 006674  000137  007604                  JMP     CMNSTART                ;JUMP TO COMMON START
115
```

```
                                .SBTTL  STANDALONE INPUT ROUTINES
                        
006700                          STANDALONE:
006700  004737  064120                  JSR     PC,$TKINT       ;INITIALIZE CONSOLE
                        
006704  005227  177777                  INC     #-1             ;FIRST TIME THRU HERE ?
006710  001426                          BEQ     3$              ;YES !!
                        
                                ;SEE IF THE USER WANTS TO KEEP SAME DEVICES FOR TESTING
006712                          1$:
006712  104401  066304                  TYPE    ,CNSL00         ;MAINTAIN PREVIOUS PARAMETERS ?
006716  104411                          RDCHR                   ;GET RESPONSE
006720  012637  001176                  MOV     (SP)+,$TMP1     ;ECHO RESPONSE
006724  104401  001176                  TYPE    ,$TMP1
006730  123727  001176  000131          CMPB    $TMP1,#'Y       ;YES RESPONSE ?
006736  001004                          BNE     2$              ;NO!!
006740  104401  001217                  TYPE    ,$CRLF          ;CR-LF
006744  000137  007604                  JMP     CMNSTART        ;KEEP PREVIOUS PARAMETERS
                        
006750  123727  001176  000116  2$:     CMPB    $TMP1,#'N       ;NO RESPONSE ?
006756  001427                          BEQ     5$              ;YES, GET NEW PARAMETERS
006760  104401  066701                  TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT '
006764  000752                          BR      1$              ;TRY AGAIN
                        
                                ;SEE IF OPERATOR WANTS HELP TEXT
006766                          3$:
006766  104401  066220                  TYPE    ,MSHELP         ;WANT HELP ?
006772  104411                          RDCHR                   ;GET RESPONSE
006774  012637  001176                  MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
007000  104401  001176                  TYPE    ,$TMP1
007004  123727  001176  000131          CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ?
007012  001407                          BEQ     4$              ;YES
007014  123727  001176  000116          CMPB    $TMP1,#'N       ;WAS IT A NO RESPONSE ?
007022  001405                          BEQ     5$              ;YES
007024  104401  066701                  TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT'
007030  000756                          BR      3$              ;TRY AGAIN
007032  104401  103640          4$:     TYPE    ,HELP           ;YES - TYPE HELP TEXT
                        
                                ;SEE IF USER WANTS TO CHANGE ADDRESSES
007036                          5$:
007036  104401  001217                  TYPE    ,$CRLF          ;CR-LF
007042  104401  066251                  TYPE    ,UBUSQST        ;WANT TO CHANGE ADDRESS ?
007046  104411                          RDCHR                   ;GET RESPONSE
007050  012637  001176                  MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
007054  104401  001176                  TYPE    ,$TMP1
007060  123727  001176  000131          CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ?
007066  001407                          BEQ     6$              ;YES
007070  123727  001176  000116          CMPB    $TMP1,#'N       ;WAS IT A NO RESPONSE ?
007076  001525                          BEQ     12$             ;YES
007100  104401  066701                  TYPE    ,CNSL08         ;NO, TYPE ' ILLEGAL INPUT '
007104  000756                          BR      5$+4            ;TRY AGAIN
                        
                                ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, VECTOR ADDRESS AND INTERRUPT PRIORITY
007106                          6$:
007106  104401  066337                  TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
007112  013746  001276                  MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
007116  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

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

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

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

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

```
114 007364  104401  001217          13$:    TYPE    ,$CRLF          ;CR-LF
115 007370  005037  001300          14$:    CLR     $DEVM           ;CLEAR DEVICE MAP
116 007374  104401  066723                  TYPE    ,MSDRVS         ;TYPE 'DRIVE(S): '
117 007400  104411                          RDCHR
118 007402  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
119 007406  023727  001176  000101          CMP     $TMP1,#'A       ;IS INPUT "A" ?
120 007414  001007                          BNE     15$             ;NO
121 007416  104401  066204                  TYPE    ,ALL            ;YES, TYPE "ALL" AND GO
122 007422  012737  000377  001300          MOV     #377,$DEVM      ;SET DEVICE MAP FOR ALL DRIVES
123 007430  000137  006454                  JMP     XSIZ            ;AUTO SIZE.
124
125 007434  023727  001176  000015  15$:    CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
126 007442  001436                          BEQ     17$             ;YES
127 007444  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
128 007450  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
129 007456  002430                          BLT     17$             ;YES
130 007460  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
131 007466  003427                          BLE     18$             ;NO
132 007470  000423                          BR      17$             ;ILLEGAL INPUT
133
134 007472  104411          16$:    RDCHR
135 007474  012637  001176                  MOV     (SP)+,$TMP1     ;GET RESPONSE
136 007500  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN ?
137 007506  001432                          BEQ     19$             ;YES
138 007510  104401  066215                  TYPE    ,COMMA          ;TYPE ', '
139 007514  104401  001176                  TYPE    ,$TMP1          ;ECHO RESPONSE
140 007520  023727  001176  000060          CMP     $TMP1,#'0       ;NUMBER < 0 ?
141 007526  002404                          BLT     17$             ;YES
142 007530  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7 ?
143 007536  003403                          BLE     18$             ;NO
144 007540  104401  066701          17$:    TYPE    ,CNSL08         ;TYPE " ?ILLEGAL INPUT"
145 007544  000711                          BR      14$             ;RETRY
146
147 007546  013701  001176          18$:    MOV     $TMP1,R1        ;R1 = DRIVE NUMBER
148 007552  042701  177770                  BIC     #^C7,R1
149 007556  156137  067146  001300          BISB    ATNTBL(R1),$DEVM        ;SET DEVICE IN MAP
150 007564  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ?
151 007572  101337                          BHI     16$             ;NO
152 007574  104401  001217          19$:    TYPE    ,$CRLF          ;CR-LF
153 007600  000137  006454                  JMP     XSIZ            ;GO SIZE DEVICES
154
```

```
                                        ;ASSEMBLE TEST QUE FROM DEVICE MAP
007604                                  CMNSTART:
007604  013700  001300                          MOV     $DEVM,R0        ;R0 = DEVICE MAP
007610  012701  001466                          MOV     #TSTQUE+2,R1    ;R1 = ADDRESS OF FIRST ENTRY IN QUE
007614  010137  001464                          MOV     R1,TSTQUE       ;INITIALIZE ENTRY POINTER
007620  012702  000001                          MOV     #1,R2           ;R2 = DEVICE POINTER
007624  005003                                  CLR     R3              ;R3 = DEVICE NUMBER
007626  030200                          1$:     BIT     R2,R0           ;IS THIS DEVICE IN MAP ?
007630  001406                                  BEQ     2$              ;NO !!
007632  010311                                  MOV     R3,(R1)         ;YES - ENTER DEVICE NUMBER IN QUE
007634  116361  067146  000001                  MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
007642  062701  000002                          ADD     #2,R1           ;ADVANCE ENTRY POINTER
007646  006302                          2$:     ASL     R2              ;ADVANCE DEVICE POINTER
007650  105702                                  TSTB    R2              ;DONE ALL DEVICES ?
007652  001402                                  BEQ     3$              ;YES
007654  005203                                  INC     R3              ;ADVANCE DEVICE NUMBER
007656  000763                                  BR      1$              ;ENTER NEXT DEVICE
007660  005011                          3$:     CLR     (R1)            ;TERMINATE TEST QUE

                                        ;SIZE FOR CLOCK
007662  004737  044644                          JSR     PC,SIZCLK       ;SEE IF CLOCK PRESENT
007666  000413                                  BR      5$              ;YES - CLOCK IS PRESENT
007670                                  4$:
007670  104000                                  EMT
007672  104401  007700                          TYPE    ,65$            ;;TYPE ASCIZ STRING
007676  000405                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF>/PROG HLT/
007712                                  64$:
007712  000000                                  HALT                    ;PROGRAM HALT !!
007714  000765                                  BR      4$
007716                                  5$:

                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
007716  005737  000042                          TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
007722  001012                                  BNE     66$             ;;BRANCH IF YES
007724  123727  001242  000001                  CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
007732  001406                                  BEQ     66$             ;;BRANCH IF YES
007734  023727  001154  000176                  CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
007742  001005                                  BNE     67$             ;;BRANCH IF NO
007744  104407                                  GTSWR                   ;;GET SOFT-SWR SETTINGS
007746  000403                                  BR      67$
007750  112737  000001  001150          66$:    MOVB    #1,$AUTOB       ;;SE. AUTO-MODE INDICATOR
007756                                  67$:

                                        READY:  NOP                     ;READY TO START TEST
007760  105737  001300                          TSTB    $DEVM           ;ANY DRIVES IN MAP ?
007764  001007                                  BNE     2$              ;BR IF YES
007766  005737  000042                          TST     @#42            ;ANY MONITOR PRESENT ?
007772  001002                                  BNE     1$              ;BR IF YES
007774  000137  005412                          JMP     START           ;JUMP TO START
010000  000137  042136          1$:     JMP     $EOP                    ;RETURN CONTROL TO MONITOR

010004  105037  001116          2$:     CLRB    $TSTNM                  ;RESET TEST NUMBER
010010  005037  001206                  CLR     $TIMES                  ;INITIALIZE NUMBER OF ITERATIONS
010014  005037  001326                  CLR     CTLFG                   ;CLEAR CONTROL-C FLAG
010020  004737  064120                  JSR     PC,$TKINT               ;INITIALIZE TTY
010024  012746  000300                  MOV     #PR6,-(SP)              ;;PUT NEW PS ON STACK
010030  012746  010036                  MOV     #64$,-(SP)              ;;PUT NEW PC ON  STACK
```

```
        010034  000002                          RTI                     ;;POP NEW PC AND PS
        010036                          64$:
     42 010036  117737  171422  001234          MOVB    @TSTQUE,$UNIT   ;LOAD UNIT NUMBER
     43
     44                                 ;CLEAR MASSBUS CONTROLLER, SELECT DRIVE AND DETERMINE THE LAST TRACK
     45                                 ;OF THE DIFFERENT DRIVE TYPES
     46 010044  012737  002000  001332          MOV     #TA4,LSTRK      ;ASSUME LAST TRACK FOR RM02/3 = 4.
     47 010052  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     48 010056  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
     49 010064  117760  171374  000010          MOVB    @TSTQUE,RMCS2(R0)       ;SELECT DEVICE UNDER TEST
     50 010072  016002  000026                  MOV     RMDT(R0),R2     ;GET RMDT AND
     51 010076  042702  177770                  BIC     #177770,R2      ;SAVE DRIVE TYPE BITS
     52 010102  022702  000007                  CMP     #7,R2           ;IS IT AN RM05 ?
     53 010106  001003                          BNE     3$              ;NO, MUST BE AN RM02 OR RM03
     54 010110  012737  011000  001332          MOV     #TA16.TA2,LSTRK ;YES--SET LAST TRACK = 18.
     55 010116                          3$:
```

```
                                ;;*****************************************************************
                                ;*TEST 1         CONTROLLER ACCESS TEST
                                ;;*****************************************************************
                                TST1:
 010116                                 SCOPE                   ;SCOPE CALL
 010116  000004                         NOP                     ;START OF TEST
 010120  000240                         MOV     #STACK,SP       ;INITIALIZE STACK POINTER
 010122  012706  001100                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
 010126  013700  001276                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
 010132  013701  001464                 MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 010136  012737  000001  001226
  3
  4 010144  005001                      CLR     R1
  5 010146  013746  000004              MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
  6 010152  013746  000006              MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
  7 010156  012737  010250  000004      MOV     #3$,ERRVEC
  8 010164  012737  000300  000006      MOV     #PR6,ERRVEC+2
  9
 10 010172  110160  000001              MOVB    R1,RMCS1+1(R0)  ;MOVE HI BYTE TO RMCS1
 11 010176  010160  000002              MOV     R1,RMWC(R0)     ;MOVE WORD COUNT REGISTER
 12 010202  016002  000002              MOV     RMWC(R0),R2
 13 010206  010160  000004              MOV     R1,RMBA(R0)     ;MOVE BUS ADDRESS REGISTER
 14 010212  016002  000004              MOV     RMBA(R0),R2
 15 010216  010160  000010              MOV     R1,RMCS2(R0)    ;MOVE CONTROL STATUS REGISTER
 16 010222  016002  000010              MOV     RMCS2(R0),R2
 17 010226  010160  000022              MOV     R1,RMDB(R0)     ;MOVE DATA BUFFER
 18 010232  016002  000022              MOV     RMDB(R0),R2
 19 010236  012637  000006              MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
 20 010242  012637  000004              MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
 21 010246  000415                      BR      7$              ;NO BUS TIMEOUT OCCURRED
 22
 23 010250  022626                      3$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
 24 010252  012637  000006              MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
 25 010256  012637  000004              MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
 26 010262  104110                      EMT     110
 27 010264  005737  000042              TST     @#42            ;STAND ALONE MODE ?
 28 010270  001002                      BNE     5$              ;NO!!
 29 010272  000137  005412              JMP     START           ;YES-GO GET $BASE
 30
 31 010276  000137  042136              5$:     JMP     $EOP            ;GO TO END OF PASS HANDLER
 32 010302                              7$:
 33
 34                                     ;;*****************************************************************
                                        ;*TEST 2         DEVICE AVAILABLE TEST
                                        ;;*****************************************************************
                                        TST2:
 010302                                 SCOPE                   ;SCOPE CALL
 010302  000004                         NOP                     ;START OF TEST
 010304  000240                         MOV     #STACK,SP       ;INITIALIZE STACK POINTER
 010306  012706  001100                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
 010312  013700  001276                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
 010316  013701  001464                 MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 010322  012737  000002  001226
 35
 36 010330  004737  053530              JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
 010334  000404                         BR      2$              ;GO TO 2$ IF NO ERROR
 010336  000240                         NOP                     ;RETURN HERE IF ERROR
 010340  104000                         EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
```

```
    010342  000137  010462                  JMP     7$              ;GO TO 7$ IF ERROR
 37 010346                          2$:
 38 010346  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
 39 010352  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
 40 010356  012737  010446  000004          MOV     #5$,ERRVEC
 41 010364  013737  000300  000006          MOV     PR6,ERRVEC+2
 42
 43 010372  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET DVA STATUS
 44 010400  016037  000010  001174          MOV     RMCS2(R0),$TMP0 ;GET NED STATUS
 45 010406  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
 46 010412  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
 47 010416  032737  010000  001174          BIT     #NED,$TMP0      ;NONEXISTENT DEVICE ?
 48 010424  001402                          BEQ     3$              ;NO!!
 49 010426  104111                          EMT     111
 50 010430  000414                          BR      7$
 51 010432  032737  004000  001176  3$:     BIT     #DVA,$TMP1      ;DEVICE AVAILABLE ?
 52 010440  001012                          BNE     9$              ;YES!!
 53 010442  104112                          EMT     112
 54 010444  000406                          BR      7$
 55
 56 010446  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
 57 010450  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
 58 010454  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
 59 010460  104113                          EMT     113
 60 010462  000137  042100          7$:     JMP     $EOSP
 61 010466                          9$:
 62
 63                                 ;;*******************************************************************
                                    ;*TEST 3         DRIVE TYPE TEST
                                    ;;*******************************************************************
    010466                          TST3:
    010466  000004                          SCOPE                   ;SCOPE CALL
    010470  000240                          NOP                     ;START OF TEST
    010472  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
    010476  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    010502  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
    010506  012737  000003  001226          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 64
 65 010514  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
    010520  000404                          BR      2$              ;GO TO 2$ IF NO ERROR
    010522  000240                          NOP                     ;RETURN HERE IF ERROR
    010524  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
    010526  000137  010664                  JMP     4$              ;GO TO 4$ IF ERROR
 66 010532                          2$:
 67 010532  112737  000026  001514          MOVB    #RMDT,GETINX    ;SETUP GET INDEX TABLE
    010540  112737  000200  001515          MOVB    #200,GETINX+1   ;SETUP TERMINATOR BYTE
    010546  012737  010670  001362          MOV     #5$,RMDTI       ;SET RMDT INPUT BUFFER = 5$
    010554  004737  044154                  JSR     PC,GET          ;GO READ RMDT VIA GET SUBROUTINE
    010560  000402                          BR      3$              ;GO TO 3$ IF NO ERROR
    010562  000240                          NOP                     ;RETURN HERE IF ERROR
    010564  104000                          EMT                     ;ERROR DEFINED BY GET SUBROUTINE
 68
 69 010566  022737  020024  001362  3$:     CMP     #SNGPRT,RMDTI   ;SINGLE PORT RM03 ?
 70 010574  001435                          BEQ     5$              ;YES !!
 71 010576  022737  024024  001362          CMP     #DULPRT,RMDTI   ;DUAL PORT RM03 ?
 72 010604  001431                          BEQ     5$              ;YES !!
 73
```

```
 74 010606  022737  020025  001362          CMP     #SNGPRT!BIT0,RMDTI      ;SINGLE PORT RM02 ?
 75 010614  001425                           BEQ     5$                      ;YES !!
 76 010616  022737  024025  001362          CMP     #DULPRT!BIT0,RMDTI      ;DUAL PORT RM02 ?
 77 010624  001421                           BEQ     5$                      ;YES !!
 78
 79 010626  022737  020027  001362          CMP     #SNGPRT!BIT1!BIT0,RMDTI ;SINGLE PORT RM05 ?
 80 010634  001415                           BEQ     5$                      ;YES !!
 81 010636  022737  024027  001362          CMP     #DULPRT!BIT1!BIT0,RMDTI ;DUAL PORT RM05 ?
 82 010644  001411                           BEQ     5$
 83
 84 010646  012737  020024  001176          MOV     #SNGPRT,$TMP1   ;LOAD ACCEPTABLE VALUES
 85 010654  012737  024024  001200          MOV     #DULPRT,$TMP2
 86 010662  104114                           EMT     114
 87
 88 010664  000137  042100          4$:     JMP     $EOSP           ;GO TO SUBPASS HANDLER.
 89 010670                          5$:
 90
 91                                 ;;*****************************************************************
                                    ;*TEST 4         UNIBUS INITIALIZE TEST
                                    ;;*****************************************************************
    010670                          TST4:
    010670  000004                          SCOPE                   ;SCOPE CALL
    010672  000240                          NOP                     ;START OF TEST
    010674  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
    010700  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    010704  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
    010710  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
    010716  012737  000004  001226          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 92
 93 010724  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
    010730  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
    010732  000240                          NOP                     ;RETURN HERE IF ERROR
    010734  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
    010736  000137  011540                  JMP     220$            ;GO TO 220$ IF ERROR
 94 010742                          10$:
 95 010742  012702  000101                  MOV     #65.,R2         ;SET OR AND RESET IR
 96 010746  012737  000000  001432          MOV     #0,RMDBO        ;WRITE ZEROS IN DATA SILO
 97 010754  112737  000022  001543          MOVB    #RMDB,PUTINX    ;SETUP REGISTER INDEX
 98 010762  112737  000200  001544          MOVB    #200,PUTINX+1
 99 010770                          20$:
100 010770  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    010774  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
    010776  000240                          NOP                     ;RETURN HERE IF ERROR
    011000  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    011002  000137  011540                  JMP     220$            ;GO TO 220$ IF ERROR
101 011006  005302                  30$:    DEC     R2              ;DECREMENT COUNT
102 011010  001367                          BNE     20$             ;WRITE SILO AGAIN
103 011012  012737  177777  001414          MOV     #-1,RMBA0       ;RMBA = ALL 1'S
104 011020  012737  177777  001424          MOV     #-1,RMER10      ;RMER1 = ALL 1'S
105 011026  012737  177777  001452          MOV     #-1,RMER20      ;RMER2 = ALL 1'S
106 011034  012737  040001  001434          MOV     #DMD!DBEN,RMMR10        ;SET DIAGNOSTIC MODE
107 011042  012737  003577  001410          MOV     #003577,RMCS10
108 011050  012737  021037  001420          MOV     #021037,RMCS20
109
110 011056  012702  001543                  MOV     #PUTINX,R2      ;R2 = ADDRESS OF INDEX TABLE
111 011062  112722  000004                  MOVB    #RMBA,(R2)+
112 011066  112722  000014                  MOVB    #RMER1,(R2)+
```

```
113 011072  112722  000042                  MOVB    #RMER2,(R2)+
114 011076  112722  000024                  MOVB    #RMMR1,(R2)+
115 011102  112722  000000                  MOVB    #RMCS1,(R2)+
116 011106  112722  000010                  MOVB    #RMCS2,(R2)+
117 011112  112722  000200                  MOVB    #200,(R2)+
118 011116  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
    011122  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
    011124  000240                          NOP                     ;RETURN HERE IF ERROR
    011126  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
    011130  000137  011540                  JMP     220$            ;GO TO 220$ IF ERROR
119 011134  000005              40$:        RESET                   ;UNIBUS INITIALIZE
120 011136  004737  064120                  JSR     PC,$TKINT       ;INITIALIZE CONSOLE
121 011142  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT DEVICE
122 011146  004737  044070                  JSR     PC,GETSTS       ;GO SET UP FOR STATUS FETCH
123
124 011152  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    011156  000403                          BR      50$             ;GO TO 50$ IF NO ERROR
    011160  000240                          NOP                     ;RETURN HERE IF ERROR
    011162  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
125 011164  000565                          BR      220$            ;SKIP REMAINDER OF TEST
126
127 011166  013737  001334  001142  50$:    MOV     RMCS11,$BDDAT   ;VERIFY RMCS1
128 011174  042737  100000  001142          BIC     #SC,$BDDAT      ;IGNORE SPECIAL CONDITION
129 011202  012737  004200  001140          MOV     #DVA!RDY,$GDDAT ;EXPECT DVA & RDY
130 011210  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED, RECEIVED
131 011216  001401                          BEQ     60$             ;BRANCH IF EQUAL
132 011220  104115                          EMT     115
133
134 011222  005037  001140          60$:    CLR     $GDDAT          ;VERIFY RMBA IS ZERO
135 011226  013737  001340  001142          MOV     RMBAI,$BDDAT
136 011234  001401                          BEQ     70$             ;BRANCH IF ZERO
137 011236  104116                          EMT     116
138
139 011240  013737  001344  001142  70$:    MOV     RMCS2I,$BDDAT   ;VERIFY RMCS2
140 011246  005046                          CLR     -(SP)           ;EXPECT IR & UNIT NUMBER
141 011250  111116                          MOVB    (R1),(SP)
142 011252  052716  000100                  BIS     #IR,(SP)
143 011256  012637  001140                  MOV     (SP)+,$GDDAT
144 011262  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
145 011270  001401                          BEQ     90$             ;BRANCH IF EQUAL
146 011272  104117                          EMT     117
147
148 011274  005037  001140          90$:    CLR     $GDDAT          ;VERIFY RMER1
149 011300  013737  001350  001142          MOV     RMER1I,$BDDAT
150 011306  042737  040000  001142          BIC     #UNS,$BDDAT     ;IGNORE UNSAFE
151 011314  001401                          BEQ     110$            ;BRANCH IF ZERO
152 011316  104120                          EMT     120
153
154 011320  013737  001352  001142  110$:   MOV     RMASI,$BDDAT    ;VERIFY RMAS
155 011326  005002                          CLR     R2              ;CLEAR ALL BUT THIS
156 011330  116102  000001                  MOVB    1(R1),R2        ;DRIVES ATTENTION BIT
157 011334  000302                          SWAB    R2
158 011336  005102                          COM     R2
159 011340  000240                          NOP
160 011342  040237  001142                  BIC     R2,$BDDAT
161 011346  001401                          BEQ     130$            ;BRANCH IF ITS 0
162 011350  104121                          EMT     121
```

```
011352  013737  001360  001142  130$:   MOV     RMMR1I,$BDDAT   ;VERIFY RMMR
011360  042737  000046  001142          BIC     #WC!LS!LST,$BDDAT       ;IGNORE WORD CLOCK, SCT, TRK
011366  012737  000010  001140          MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA BIT
011374  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
011402  001401                          BEQ     170$            ;BRANCH IF 0
011404  104122                          EMT     122

011406  005037  001140          170$:   CLR     $GDDAT          ;EXPECT ZEROS
011412  013737  001402  001142          MOV     RMEC2I,$BDDAT   ;VERIFY RMEC2 = 0
011420  001401                          BEQ     190$
011422  104124                          EMT     124

011424  013737  001374  001142  190$:   MOV     RMMR2I,$BDDAT   ;VERIFY RMMR2
011432  042737  140000  001142          BIC     #RQA!RQB,$BDDAT
011440  012737  011777  001140          MOV     #TST!1777,$GDDAT        ;EXPECT TEST,TAG BIT ON
011446  023737  001140  001142          CMP     $GDDAT,$BDDAT
011454  001401                          BEQ     210$
011456  104125                          EMT     125

011460  005037  001140          210$:   CLR     $GDDAT          ;EXPECT ALL ZEROS
011464  013737  001376  001142          MOV     RMER2I,$BDDAT   ;VERIFY RMER2
011472  042737  040200  001142          BIC     #SKI!DVC,$BDDAT ;IGNORE DEVICE ERRORS
011500  001401                          BEQ     215$            ;BRANCH IF OTHER BITS 0
011502  104173                          EMT     173
011504  013737  001346  001142  215$:   MOV     RMDSI,$BDDAT    ;CHECK DRIVE STATUS REGISTER
011512  042737  177177  001142          BIC     #^C<DPR!DRY>,$BDDAT
011520  012737  000600  001140          MOV     #DPR!DRY,$GDDAT ;EXPECTED STATUS
011526  023737  001142  001140          CMP     $BDDAT,$GDDAT   ;COMPARE EXPECTED & RECEIVED STATUS
011534  001401                          BEQ     220$            ;DRIVE STATUS IS OK
011536  104123                          EMT     123
011540                          220$:


                                ;;*****************************************************************
                                ;*TEST 5        CONTROLLER CLEAR TEST
                                ;;*****************************************************************
011540                          TST5:
011540  000004                          SCOPE                   ;SCOPE CALL
011542  000240                          NOP                     ;START OF TEST
011544  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
011550  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
011554  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
011560  012737  000005  001226          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

011566  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
011572  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
011574  000240                          NOP                     ;RETURN HERE IF ERROR
011576  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
011600  000137  012076                  JMP     80$             ;GO TO 80$ IF ERROR
011604                          10$:
011604  012702  000101                  MOV     #65.,R2
011610  012737  000000  001432          MOV     #0,RMDB0        ;WRITE ZEROS IN DATA SILO
011616  112737  000022  001543          MOVB    #RMDB,PUTINX    ;SETUP REGISTER INDEX TABLE
011624  112737  000200  001544          MOVB    #200,PUTINX+1
011632                          20$:
011632  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
011636  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
```

```
        011640  000240                          NOP                     ;RETURN HERE IF ERROR
        011642  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        011644  000137  012076                  JMP     80$             ;GO TO 80$ IF ERROR
   206  011650  005302                  30$:    DEC     R2              ;DECREMENT COUNT
   207  011652  001367                          BNE     20$             ;AND WRITE SILO AGAIN IF NOT DONE
   208  011654  012737  177777  001414          MOV     #-1,RMBA0
   209  011662  012737  177777  001424          MOV     #-1,RMER10
   210  011670  012737  177777  001452          MOV     #-1,RMER20
   211  011676  012737  040001  001434          MOV     #DMD!DBEN,RMMR10
   212  011704  012737  003577  001410          MOV     #003577,RMCS10
   213  011712  012737  021037  001420          MOV     #021037,RMCS20
   214  011720  012702  001543                  MOV     #PUTINX,R2      ;R2 = ADDRESS OF INDEX TABLE
   215  011724  112722  000004                  MOVB    #RMBA,(R2)+
   216  011730  112722  000014                  MOVB    #RMER1,(R2)+
   217  011734  112722  000042                  MOVB    #RMER2,(R2)+
   218  011740  112722  000024                  MOVB    #RMMR1,(R2)+
   219  011744  112722  000000                  MOVB    #RMCS1,(R2)+
   220  011750  112722  000010                  MOVB    #RMCS2,(R2)+
   221  011754  112722  000200                  MOVB    #200,(R2)+
   222  011760  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        011764  000403                          BR      40$             ;GO TO 40$ IF NO ERROR
        011766  000240                          NOP                     ;RETURN HERE IF ERROR
        011770  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
   223  011772  000430                          BR      70$
   224
   225  011774                          40$:
   226  011774  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
        012000  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        012002  000240                          NOP                     ;RETURN HERE IF ERROR
        012004  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
        012006  000137  012076                  JMP     80$             ;GO TO 80$ IF ERROR
   227  012012  004737  044070          50$:    JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
   228
   229  012016  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        012022  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
        012024  000240                          NOP                     ;RETURN HERE IF ERROR
        012026  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        012030  000137  012076                  JMP     80$             ;GO TO 80$ IF ERROR
   230  012034                          60$:
   231  012034  004737  053646                  JSR     PC,CLRSTS       ;GO VERIFY CONTROLLER CLEAR OPERATION
        012040  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        012042  000240                          NOP                     ;RETURN HERE IF ERROR
        012044  104000                          EMT                     ;ERROR # DEFINED BY CLRSTS SUBROUTINE
        012046  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        012050  000137  012076                  JMP     80$             ;GO TO 80$ IF ERROR
   232  012054                          70$:
   233  012054  012737  000000  001410          MOV     #NOP,RMCS10     ;CHANGE  FUNCTION CODE FOR ERROR CHECK
   234
   235  012062  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        012066  000403                          BR      80$             ;GO TO 80$ IF NO ERROR
        012070  000240                          NOP                     ;RETURN HERE IF ERROR
        012072  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        012074  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
   236  012076                          80$:
   237
   238                                  ;;*****************************************************************
                                        ;*TEST 6         ERROR CLEAR TEST
```

```
                                        ;;*****************************************************************
                                        ;*TEST 6        ERROR CLEAR TEST
                                        ;;*****************************************************************
012076                                  TST6:
012076  000004                                  SCOPE                   ;SCOPE CALL
012100  000240                                  NOP                     ;START OF TEST
012102  012706  001100                          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
012106  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
012112  013701  001464                          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
012116  012737  000006  001226                  MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
239
240 012124  004737  053530                      JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
012130  000404                                  BR      10$             ;GO TO 10$ IF NO ERROR
012132  000240                                  NOP                     ;RETURN HERE IF ERROR
012134  104000                                  EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
012136  000137  012302                          JMP     50$             ;GO TO 50$ IF ERROR
241 012142                              10$:
242 012142  112737  000000  001543              MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
012150  112737  000200  001544                  MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
012156  012737  040000  001410                  MOV     #TRE,RMCS10     ;SET RMCS1 OUTPUT BUFFER = TRE
012164  004737  044424                          JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
012170  000403                                  BR      20$             ;GO TO 20$ IF NO ERROR
012172  000240                                  NOP                     ;RETURN HERE IF ERROR
012174  104000                                  EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
243 012176  000441                              BR      50$             ;SKIP REST OF TEST
244 012200  112737  000000  001514  20$:         MOVB    #RMCS1,GETINX
245 012206  112737  000010  001515              MOVB    #RMCS2,GETINX+1
246 012214  112737  000200  001516              MOVB    #200,GETINX+2
247
248 012222  004737  044154                      JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012226  000403                                  BR      30$             ;GO TO 30$ IF NO ERROR
012230  000240                                  NOP                     ;RETURN HERE IF ERROR
012232  104000                                  EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
249 012234  000422                              BR      50$             ;SKIP REST OF TEST
250 012236  013737  001334  001142  30$:         MOV     RMCS11,$BDDAT    ;CHECK TRE & MCPE
251 012244  005037  001140                      CLR     $GDDAT          ;EXPECT 0'S
252 012250  042737  117777  001142              BIC     #^C<TRE!MCPE>,$BDDAT
253 012256  001401                              BEQ     40$             ;BRANCH IF TRE & MCPE = 0
254 012260  104137                              EMT     137
255
256 012262  013737  001344  001142  40$:         MOV     RMCS21,$BDDAT  ;CHECK RMCS2
257 012270  042737  104377  001142              BIC     #^C<WCE!UPE!NED!PGE!MXF!MDPE>,$BDDAT
258 012276  001401                              BEQ     50$
259 012300  104140                              EMT     140
260
261 012302                              50$:
262
263                                     ;;*****************************************************************
                                        ;*TEST 7        DRIVE STATUS TEST
                                        ;;*****************************************************************
012302                                  TST7:
012302  000004                                  SCOPE                   ;SCOPE CALL
012304  000240                                  NOP                     ;START OF TEST
012306  012706  001100                          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
012312  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
012316  013701  001464                          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
012322  012737  000007  001226                  MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
264
265 012330  004737  053530                      JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
```

```
        012334  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
        012336  000240                          NOP                     ;RETURN HERE IF ERROR
        012340  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
        012342  000137  012616                  JMP     100$            ;GO TO 100$ IF ERROR
   266  012346                          10$:
   267  012346  004737  044070                JSR     PC,GETSTS
   268
   269  012352  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        012356  000403                          BR      20$             ;GO TO 20$ IF NO ERROR
        012360  000240                          NOP                     ;RETURN HERE IF ERROR
        012362  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
   270  012364  000514                          BR      100$            ;SKIP REST OF TEST
   271  012366  032737  010000  001346  20$:    BIT     #MOL,RMDSI      ;MEDIUM ON LINE??
   272  012374  001016                          BNE     30$             ;YES!!
   273  012376  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
   274  012404  042737  162000  001140          BIC     #ATA!ERR!PIP!LBT,$GDDAT
   275  012412  052737  010000  001140          BIS     #MOL,$GDDAT
   276  012420  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
   277  012426  104151                          EMT     151             ;MOL STATUS INCORRECT
   278  012430  000440                          BR      70$
   279
   280  012432  032737  000200  001376  30$:    BIT     #DVC,RMER2I     ;IS THERE A DEVICE CHECK??
   281  012440  001422                          BEQ     60$             ;BR IF NO
   282  012442  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
   283  012450  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
   284  012454  104152                          EMT     152             ;DRIVE FAULT
   285
   286  012456  032737  040000  001350          BIT     #UNS,RMER1I     ;IS UNS SET??
   287  012464  001022                          BNE     70$             ;YES!!
   288  012466  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
   289  012474  012737  040000  001140          MOV     #UNS,$GDDAT     ;EXPECTED STATUS
   290  012502  104153                          EMT     153             ;"DVC" IS SET BUT "UNS" IS NOT
   291  012504  000412                          BR      70$
   292
   293  012506  032737  040000  001350  60$:    BIT     #UNS,RMER1I     ;IS UNS SET??
   294  012514  001410                          BEQ     80$             ;NO!!
   295  012516  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
   296  012524  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
   297  012530  104154                          EMT     154             ;"DVC" IS NOT SET, BUT "UNS" IS
   298  012532  000137  042100          70$:    JMP     $EOSP           ;SKIP REST OF TEST
   299
   300  012536  032737  004000  001346  80$:    BIT     #WRL,RMDSI      ;IS WRITE PROTECT ON??
   301  012544  001412                          BEQ     90$             ;NO!!
   302  012546  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
   303  012554  042737  166076  001140          BIC     #^C<MOL!PGM!DPR!DRY!VV!OM>,$GDDAT
   304  012562  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
   305  012570  104175                          EMT     175             ;SELECTED DEVICE IN WRITE PROTECT
   306
   307  012572  032737  040000  001376  90$:    BIT     #SKI,RMER2I     ;IS SKI SET??
   308  012600  001406                          BEQ     100$            ;NO!!
   309  012602  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
   310  012610  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
   311  012614  104205                          EMT     205             ;PERSISTENT SEEK INCOMPLETE ERROR
   312  012616                          100$:
   313
   314                                  ;;*****************************************************************
                                        ;*TEST 10        PRIMARY/SECONDARY ERROR TEST
```

```
                                    ;;*****************************************************************
                                    TST10:
012616                              
012616  000004                      SCOPE                   ;SCOPE CALL
012620  000240                      NOP                     ;START OF TEST
012622  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
012626  013700  001276              MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
012632  013701  001464              MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
012636  012737  000010  001226      MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
315
316 012644  004737  053530          JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
012650  000404                      BR      1$              ;GO TO 1$ IF NO ERROR
012652  000240                      NOP                     ;RETURN HERE IF ERROR
012654  104000                      EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
012656  000137  012776              JMP     5$              ;GO TO 5$ IF ERROR
317 012662                          1$:
318 012662  112737  000000  001543  MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
012670  112737  000200  001544      MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
012676  012737  000023  001410      MOV     #PAKACK!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = PAKACK!GO
012704  004737  044424              JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
012710  000403                      BR      2$              ;GO TO 2$ IF NO ERROR
012712  000240                      NOP                     ;RETURN HERE IF ERROR
012714  104000                      EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
319 012716  000427                  BR      5$              ;SKIP REST OF TEST IF ERROR
320 012720  004737  044070          2$:     JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
321 012724  004737  044766          JSR     PC,T''OUT       ;WAIT FOR COMPLETION OF NOP
322
323 012730  004737  044154          JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
012734  000403                      BR      3$              ;GO TO 3$ IF NO ERROR
012736  000240                      NOP                     ;RETURN HERE IF ERROR
012740  104000                      EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
324 012742  000415                  BR      5$              ;SKIP REST OF TEST IF ERROR
325 012744                          3$:
326 012744  004737  045152          JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
012750  000404                      BR      4$              ;GO TO 4$ IF NO ERROR
012752  000240                      NOP                     ;RETURN HERE IF ERROR
012754  104000                      EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
012756  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
327 012760  000406                  BR      5$              ;SKIP REST OF TEST IF ERROR
328 012762                          4$:
329 012762  004737  046004          JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
012766  000403                      BR      5$              ;GO TO 5$ IF NO ERROR
012770  000240                      NOP                     ;RETURN HERE IF ERROR
012772  104000                      EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
012774  004736                      JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
330 012776                          5$:
331
332                                 ;;*****************************************************************
                                    ;*TEST 11        DIAGNOSTIC MODE TEST
                                    ;;*****************************************************************
012776                              TST11:
012776  000004                      SCOPE                   ;SCOPE CALL
013000  000240                      NOP                     ;START OF TEST
013002  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
013006  013700  001276              MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
013012  013701  001464              MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
013016  012737  000011  001226      MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
333
```

```
                                                        ;GO ISSUE CONTROLLER CLEAR
013024  004737  053530            JSR     PC,CNTCLR
013030  000404                    BR      1$              ;GO TO 1$ IF NO ERROR
013032  000240                    NOP                     ;RETURN HERE IF ERROR
013034  104000                    EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
013036  000137  014050            JMP     22$             ;GO TO 22$ IF ERROR
013042                    1$:
013042  112737  000024  001543    MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
013050  112737  000200  001544    MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
013056  012737  000001  001434    MOV     #DMD,RMMR1O     ;SET RMMR1 OUTPUT BUFFER = DMD
013064  004737  044424            JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
013070  000404                    BR      2$              ;GO TO 2$ IF NO ERROR
013072  000240                    NOP                     ;RETURN HERE IF ERROR
013074  104000                    EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
013076  000137  014050            JMP     22$             ;GO TO 22$ IF ERROR
013102  004737  044070    2$:     JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH

013106  004737  044154            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
013112  000404                    BR      3$              ;GO TO 3$ IF NO ERROR
013114  000240                    NOP                     ;RETURN HERE IF ERROR
013116  104000                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
013120  000137  014050            JMP     22$             ;GO TO 22$ IF ERROR
013124  032737  000001  001360  3$:     BIT     #DMD,RMMR1I     ;IS  DIAGNOSTIC MODE SET??
013132  001011                    BNE     5$              ;YES!!
013134  012737  000001  001140    MOV     #DMD,$GDDAT     ;EXPECTED STATUS
013142  013737  001360  001142    MOV     RMMR1I,$BDDAT   ;RECEIVED STATUS
013150  104176                    EMT     176
013152  000137  014050            JMP     22$             ;SKIP REST OF TEST IF DMD = 0
013156  032737  010000  001346  5$:     BIT     #MOL,RMDSI      ;IS "MOL" = 0   ??
013164  001411                    BEQ     6$              ;YES!!
013166  013737  001346  001142    MOV     RMDSI,$BDDAT    ;SETUP BAD DATA FOR TYPEOUT
013174  042737  167777  001142    BIC     #^CMOL,$BDDAT
013202  005037  001140            CLR     $GDDAT          ;SETUP GOOD DATA FOR TYPEOUT
013206  104177                    EMT     177
013210  032737  020000  001346  6$:     BIT     #PIP,RMDSI      ;IS PIP  SET??
013216  001012                    BNE     7$              ;YES!!
013220  013737  001346  001142    MOV     RMDSI,$BDDAT    ;SETUP BAD DATA FOR TYPEOUT
013226  042737  157777  001142    BIC     #^CPIP,$BDDAT
013234  012737  020000  001140    MOV     #PIP,$GDDAT     ;EXPECTED PIP SET
013242  104200                    EMT     200
013244  032737  004000  001346  7$:     BIT     #WRL,RMDSI      ;IS WRITE LOCK OFF??
013252  001411                    BEQ     8$              ;YES!!
013254  013737  001346  001142    MOV     RMDSI,$BDDAT    ;SETUP BAD DATA FOR TYPEOUT
013262  042737  173777  001142    BIC     #^CWRL,$BDDAT
013270  005037  001140            CLR     $GDDAT          ;EXPECTED WRL = 0
013274  104201                    EMT     201
013276  032737  040000  001376  8$:     BIT     #SKI,RMER2I     ;IS SKI = 0
013304  001411                    BEQ     9$              ;YES.!
013306  013737  001376  001142    MOV     RMER2I,$BDDAT   ;SETUP BAD DATA FOR TYPEOUT
013314  042737  137777  001142    BIC     #^CSKI,$BDDAT
013322  005037  001140            CLR     $GDDAT          ;SKI SHOULD BE 0
013326  104202                    EMT     202
013330  032737  000200  001376  9$:     BIT     #DVC,RMER2I     ;IS DEVICE CHECK = 0??
013336  001411                    BEQ     10$             ;YES!!
013340  013737  001376  001142    MOV     RMER2I,$BDDAT   ;SETUP BAD DATA FOR  TYPEOUT
013346  042737  177577  001142    BIC     #^CDVC,$BDDAT
013354  005037  001140            CLR     $GDDAT          ;DVC SHOULD BE 0
013360  104203                    EMT     203
```

```
376 013362                          10$:
377 013362  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
    013370  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
    013376  012737  001711  C01434          MOV     #DMD.MUR!MOC!MSER!MDF!MWP,RMMR10        ;SET RMMR1 OUTPUT BUFFER = DMD MUR M
    013404  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
    013410  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
    013412  000240                          NOP                     ;RETURN HERE IF ERROR
    013414  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
    013416  000137  014050                  JMP     22$             ;GO TO 22$ IF ERROR
378 013422                          11$:
379 013422  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    013426  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
    013430  000240                          NOP                     ;RETURN HERE IF ERROR
    013432  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
    013434  000137  014050                  JMP     22$             ;GO TO 22$ IF ERROR
380 013440  032737  000001  001360  12$:    BIT     #DMD.RMMR1I     ;IS  DIAGNOSTIC MODE SET??
381 013446  001011                          BNE     125$            ;YES!!
382 013450  012737  000001  001140          MOV     #DMD,$GDDAT     ;EXPECTED STATUS
383 013456  013737  001360  001142          MOV     RMMR1I,$BDDAT   ;RECEIVED STATUS
384 013464  104176                          EMT     176
385 013466  000137  014050                  JMP     22$             ;SKIP REST OF TEST IF DMD = 0
386 013472  032737  010000  001346  125$:   BIT     #MOL,RMDSI      ;IS MOL = 1??
387 013500  001012                          BNE     13$             ;YES!!
388 013502  013737  001346  001142          MOV     RMDSI,$BDDAT    ;SETUP BAD DATA FOR TYPEOUT
389 013510  042737  167777  001142          BIC     #^CMOL,$BDDAT
390 013516  012737  010000  001140          MOV     #MOL,$GDDAT     ;EXPECTED MOL = 1
391 013524  104177                          EMT     177
392 013526  032737  020000  001346  13$:    BIT     #PIP,RMDSI      ;IS PIP = 0 ?
393 013534  001411                          BEQ     14$             ;YES!!
394 013536  013737  001346  001142          MOV     RMDSI,$BDDAT    ;SETUP BAD DATA FOR TYPEOUT
395 013544  042737  157777  001142          BIC     #^CPIP,$BDDAT
396 013552  005037  001140                  CLR     $GDDAT          ;EXPECTED PIP STATUS
397 013556  104200                          EMT     200
398 013560  032737  004000  001346  14$:    BIT     #WRL,RMDSI      ;IS WRL SET??
399 013566  001012                          BNE     15$             ;YES!!
400 013570  013737  001346  001142          MOV     RMDSI,$BDDAT    ;SETUP BAD DATA FOR TYPEOUT
401 013576  042737  173777  001142          BIC     #^CWRL,$BDDAT
402 013604  012737  004000  001140          MOV     #WRL,$GDDAT     ;EXPECTED GOOD STATUS
403 013612  104201                          EMT     201
404 013614  032737  040000  001376  15$:    BIT     #SKI,RMER2I     ;IS SKI SET??
405 013622  001012                          BNE     16$             ;YES!!
406 013624  013737  001376  001142          MOV     RMER2I,$BDDAT   ;BAD DATA FOR TYPEOUT
407 013632  042737  137777  001142          BIC     #^CSKI,$BDDAT
408 013640  012737  040000  001140          MOV     #SKI,$GDDAT     ;EXPECTED SKI  ON
409 013646  104202                          EMT     202
410 013650  032737  000200  001376  16$:    BIT     #DVC,RMER2I     ;IS DVC SET  ??
411 013656  001012                          BNE     17$             ;YES!!
412 013660  013737  001376  001142          MOV     RMER2I,$BDDAT   ;BAD DATA FOR TYPEOUT
413 013666  042737  177577  001142          BIC     #^CDVC,$BDDAT
414 013674  012737  000200  001140          MOV     #DVC,$GDDAT     ;EXPECTED DVC ON
415 013702  104203                          EMT     203
416 013704  032737  000100  001346  17$:    BIT     #VV,RMDSI       ;MUR SHOULD HAVE RESET VOLUME VALID
417 013712  001411                          BEQ     19$             ;BRANCH IF IT DID
418 013714  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
419 013722  042737  177677  001142          BIC     #^CVV,$BDDAT
420 013730  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
421 013734  104204                          EMT     204
```

```
                                19$:
                                        MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
                                        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                                        MOV     #DRVCLR!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = DRVCLR.GO
                                        JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
                                        BR      20$             ;GO TO 20$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                                        JMP     22$             ;GO TO 22$ IF ERROR
                                20$:
                                        JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                                        BR      21$             ;GO TO 21$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                                        JMP     22$             ;GO TO 22$ IF ERROR
                                21$:    BIT #LBC,RMER2I         ;IS LBC SET ??
                                        BNE 22$                 ;YES!!
                                        MOV RMER2I,$BDDAT       ;BAD DATA FOR TYPEOUT
                                        MOV #LBC, $GDDAT        ;GOOD DATA FOR TYPEOUT
                                        BIC #LBC, $BDDAT
                                        EMT     262
                                22$:

                                ;;*****************************************************************
                                ;*TEST 12       PACK ACKNOWLEDGE TEST
                                ;;*****************************************************************
                                TST12:
                                        SCOPE                   ;SCOPE CALL
                                        NOP                     ;START OF TEST
                                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                                        MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                        JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
                                        BR      1$              ;GO TO 1$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
                                        JMP     9$              ;GO TO 9$ IF ERROR
                                1$:
                                        MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
                                        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                                        MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
                                        JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
                                        BR      2$              ;GO TO 2$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                                        JMP     9$              ;GO TO 9$ IF ERROR
                                2$:
                                        MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
                                        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                                        MOV     #0,RMMR10       ;SET RMMR1 OUTPUT BUFFER = 0
                                        JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
                                        BR      3$              ;GO TO 3$ IF NO ERROR
                                        NOP                     ;RETURN HERE IF ERROR
                                        EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                                        JMP     9$              ;GO TO 9$ IF ERROR
```

```
                                        3$:
014214                                  
014214  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
014222  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
014230  012737  000023  001410          MOV     #PAKACK!GO,RMCS1O       ;SET RMCS1 OUTPUT BUFFER = PAKACK.GO
014236  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
014242  000404                          BR      4$              ;GO TO 4$ IF NO ERROR
014244  000240                          NOP                     ;RETURN HERE IF ERROR
014246  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
014250  000137  014350                  JMP     9$              ;GO TO 9$ IF ERROR
014254  004737  044070          4$:     JSR     PC,GETSTS
014260  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
                                        
014264  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
014270  000403                          BR      5$              ;GO TO 5$ IF NO ERROR
014272  000240                          NOP                     ;RETURN HERE IF ERROR
014274  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
014276  000424                          BR      9$              ;SKIP REMAINDER OF TEST
014300                                  5$:
014300  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
014304  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
014306  000240                          NOP                     ;RETURN HERE IF ERROR
014310  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
014312  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014314  000415                          BR      9$              ;SKIP TO NEXT TEST
014316                                  6$:
014316  004737  054526                  JSR     PC,ACKSTS       ;GO VERIFY PACK ACKNOWLEDGE
014322  000404                          BR      7$              ;GO TO 7$ IF NO ERROR
014324  000240                          NOP                     ;RETURN HERE IF ERROR
014326  104000                          EMT                     ;ERROR # DEFINED BY ACKSTS SUBROUTINE
014330  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014332  000406                          BR      9$              ;SKIP TO NEXT TEST
                                        
014334                                  7$:
014334  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
014340  000403                          BR      9$              ;GO TO 9$ IF NO ERROR
014342  000240                          NOP                     ;RETURN HERE IF ERROR
014344  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
014346  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
014350                                  9$:
                                        
                                        ;;*****************************************************************
                                        ;*TEST 13       RECALIBRATE TEST
                                        ;;*****************************************************************
014350                                  TST13:
014350  000004                          SCOPE                   ;SCOPE CALL
014352  000240                          NOP                     ;START OF TEST
014354  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
014360  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
014364  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
014370  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
014376  012737  000013  001226          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                        
014404  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
014410  050020                          .WORD   050020          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
```

```
        014412  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
        014414  000240                          NOP                     ;RETURN HERE IF ERROR
        014416  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        014420  000137  014566                  JMP     10$             ;GO TO 10$ IF ERROR
    462 014424                          5$:
    463 014424  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
        014432  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        014440  012737  000007  001410          MOV     #RECAL.GO,RMCS10        ;SET RMCS1 OUTPUT BUFFER = RECAL.GO
        014446  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
        014452  000404                          BR      6$              ;GO TO 6$ IF NO ERROR
        014454  000240                          NOP                     ;RETURN HERE IF ERROR
        014456  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        014460  000137  014566                  JMP     10$             ;GO TO 10$ IF ERROR
    464 014464  004737  044070          6$:     JSR     PC,GETSTS       ;GO SETUP FOR STATUS FETCH
    465 014470  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
    466
    467 014474  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        014500  000404                          BR      7$              ;GO TO 7$ IF NO ERROR
        014502  000240                          NOP                     ;RETURN HERE IF ERROR
        014504  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        014506  000137  014566                  JMP     10$             ;GO TO 10$ IF ERROR
    468 014512                          7$:
    469 014512  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        014516  000405                          BR      8$              ;GO TO 8$ IF NO ERROR
        014520  000240                          NOP                     ;RETURN HERE IF ERROR
        014522  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        014524  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014526  000137  014566                  JMP     10$             ;GO TO 10$ IF ERROR
    470 014532                          8$:
    471 014532  004737  055322                  JSR     PC,RCLSTS       ;GO VERIFY RECALIBRATE OPERATION
        014536  000405                          BR      9$              ;GO TO 9$ IF NO ERROR
        014540  000240                          NOP                     ;RETURN HERE IF ERROR
        014542  104000                          EMT                     ;ERROR # DEFINED BY RCLSTS SUBROUTINE
        014544  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        014546  000137  014566                  JMP     10$             ;GO TO 10$ IF ERROR
    472 014552                          9$.
    473 014552  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        014556  000403                          BR      10$             ;GO TO 10$ IF NO ERROR
        014560  000240                          NOP                     ;RETURN HERE IF ERROR
        014562  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        014564  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    474 014566                          10$:
    475
    476                                 ;;**************************************************************
                                        ;*TEST 14        ABORT RECALIBRATE TEST
                                        ;;**************************************************************
        014566                          TST14:
        014566  000004                          SCOPE                   ;SCOPE CALL
        014570  000240                          NOP                     ;START OF TEST
        014572  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        014576  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        014602  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        014606  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
        014614  012737  000014  001226          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    477
    478 014622  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        014626  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
```

```
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
     014630  000404                          BR      8$                 ;GO TO 8$ IF NO ERROR
     014632  000240                          NOP                        ;RETURN HERE IF ERROR
     014634  104000                          EMT                        ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     014636  000137  015074                  JMP     15$                ;GO TO 15$ IF ERROR
479  014642                          8$:
480  014642  112737  000014  001543         MOVB    #RMER1,PUTINX      ;SETUP PUT INDEX TABLE
     014650  112737  000200  001544         MOVB    #200,PUTINX+1      ;SET TERMINATOR BYTE
     014656  012737  040000  001424         MOV     #UNS,RMER10        ;SET RMER1 OUTPUT BUFFER = UNS
     014664  004737  044424                 JSR     PC,PUT             ;GO WRITE RMER1 VIA PUT SUBROUTINE
     014670  000404                          BR      9$                 ;GO TO 9$ IF NO ERROR
     014672  000240                          NOP                        ;RETURN HERE IF ERROR
     014674  104000                          EMT                        ;ERROR DEFINED BY PUT SUBROUTINE
     014676  000137  015074                  JMP     15$                ;GO TO 15$ IF ERROR
481  014702                          9$:
482  014702  112737  000000  001543         MOVB    #RMCS1,PUTINX      ;SETUP PUT INDEX TABLE
     014710  112737  000200  001544         MOVB    #200,PUTINX+1      ;SET TERMINATOR BYTE
     014716  012737  000007  001410         MOV     #RECAL!GO,RMCS10           ;SET RMCS1 OUTPUT BUFFER = RECAL!GO
     014724  004737  044424                 JSR     PC,PUT             ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     014730  000404                          BR      10$                ;GO TO 10$ IF NO ERROR
     014732  000240                          NOP                        ;RETURN HERE IF ERROR
     014734  104000                          EMT                        ;ERROR DEFINED BY PUT SUBROUTINE
     014736  000137  015074                  JMP     15$                ;GO TO 15$ IF ERROR
483  014742  004737  044070          10$:    JSR     PC,GETSTS          ;SETUP FOR STATUS FETCH
484
485  014746  004737  044154                 JSR     PC,GET             ;GO READ REGISTER(S) WITH GET SUBROUTINE
     014752  000404                          BR      11$                ;GO TO 11$ IF NO ERROR
     014754  000240                          NOP                        ;RETURN HERE IF ERROR
     014756  104000                          EMT                        ;ERROR # DEFINED BY GET SUBROUTINE
     014760  000137  015074                  JMP     15$                ;GO TO 15$ IF ERROR
486  014764  032737  020000  001346  11$:    BIT     #PIP,RMDSI         ;DID THE DRIVE RECALIBRATE??
487  014772  001401                          BEQ     12$                ;NO.!
488  014774  104214                          EMT     214
489  014776  004737  044766          12$:    JSR     PC,TIMOUT          ;WAIT FOR  COMPLETION
490
491  015002  004737  044154                 JSR     PC,GET             ;GO READ REGISTER(S) WITH GET SUBROUTINE
     015006  000404                          BR      13$                ;GO TO 13$ IF NO ERROR
     015010  000240                          NOP                        ;RETURN HERE IF ERROR
     015012  104000                          EMT                        ;ERROR # DEFINED BY GET SUBROUTINE
     015014  000137  015074                  JMP     15$                ;GO TO 15$ IF ERROR
492  015020                          13$:
493  015020  004737  045152                 JSR     PC,PRIERR          ;GO CHECK FOR PRIMARY ERRORS
     015024  000405                          BR      14$                ;GO TO 14$ IF NO ERROR
     015026  000240                          NOP                        ;RETURN HERE IF ERROR
     015030  104000                          EMT                        ;ERROR # DEFINED BY PRIERR SUBROUTINE
     015032  004736                          JSR     PC,@(SP)+          ;GO BACK FOR MORE ERROR CHECKS
     015034  000137  015074                  JMP     15$                ;GO TO 15$ IF ERROR
494  015040                          14$:
495  015040  004737  061232                 JSR     PC,STCDRVSTS       ;GO CHECK FOR CHANGES IN DRIVE STATUS
     015044  000405                          BR      141$               ;GO TO 141$ IF NO ERROR
     015046  000240                          NOP                        ;RETURN HERE IF ERROR
     015050  104000                          EMT                        ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
```

```
                                                                    ;GO BACK FOR MORE ERROR CHECKS
015052  004736                      JSR     PC,@(SP)+
015054  000137  015074              JMP     15$                     ;GO TO 15$ IF ERROR
496 015060                  141$:
497 015060  004737  046004          JSR     PC,SECERR               ;GO CHECK FOR SECONDARY ERRORS
015064  000403                      BR      15$                     ;GO TO 15$ IF NO ERROR
015066  000240                      NOP                             ;RETURN HERE IF ERROR
015070  104000                      EMT                             ;ERROR # DEFINED BY SECERR SUBROUTINE
015072  004736                      JSR     PC,@(SP)+               ;GO BACK FOR MORE ERROR CHECKS
498 015074                  15$:
499
500                                 ;;*****************************************************************
                                    ;*TEST 15       IVC RECALIBRATE TEST
                                    ;;*****************************************************************
015074                      TST15:
015074  000004                      SCOPE                           ;SCOPE CALL
015076  000240                      NOP                             ;START OF TEST
015100  012706  001100              MOV     #STACK,SP               ;INITIALIZE STACK POINTER
015104  013700  001276              MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS
015110  013701  001464              MOV     TSTQUE,R1               ;(R1) = DEVICE BEING TESTED
015114  012737  000001  001206      MOV     #1,$TIMES               ;;DO 1 ITERATION
015122  012737  000015  001226      MOV     #15,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
501
502 015130  004737  043150          JSR     PC,TSTPRP               ;PREPARE DEVICE FOR TEST
015134  054130                      .WORD   054130                  ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
015136  000404                      BR      8$                      ;GO TO 8$ IF NO ERROR
015140  000240                      NOP                             ;RETURN HERE IF ERROR
015142  104000                      EMT                             ;ERROR # DEFINED BY TSTPRP SUBROUTINE
015144  000137  015404              JMP     15$                     ;GO TO 15$ IF ERROR
503 015150                  8$:
504 015150  112737  000024  001543  MOVB    #RMMR1,PUTINX           ;SETUP PUT INDEX TABLE
015156  112737  000200  001544      MOVB    #200,PUTINX+1           ;SET TERMINATOR BYTE
015164  012737  000001  001434      MOV     #DMD,RMMR10             ;SET RMMR1 OUTPUT BUFFER = DMD
015172  004737  044424              JSR     PC,PUT                  ;GO WRITE RMMR1 VIA PUT SUBROUTINE
015176  000404                      BR      9$                      ;GO TO 9$ IF NO ERROR
015200  000240                      NOP                             ;RETURN HERE IF ERROR
015202  104000                      EMT                             ;ERROR DEFINED BY PUT SUBROUTINE
015204  000137  015404              JMP     15$                     ;GO TO 15$ IF ERROR
505 015210                  9$:
506 015210  112737  000024  001543  MOVB    #RMMR1,PUTINX           ;SETUP PUT INDEX TABLE
015216  112737  000200  001544      MOVB    #200,PUTINX+1           ;SET TERMINATOR BYTE
015224  012737  000000  001434      MOV     #0,RMMR10               ;SET RMMR1 OUTPUT BUFFER = 0
015232  004737  044424              JSR     PC,PUT                  ;GO WRITE RMMR1 VIA PUT SUBROUTINE
015236  000404                      BR      10$                     ;GO TO 10$ IF NO ERROR
015240  000240                      NOP                             ;RETURN HERE IF ERROR
015242  104000                      EMT                             ;ERROR DEFINED BY PUT SUBROUTINE
015244  000137  015404              JMP     15$                     ;GO TO 15$ IF ERROR
507 015250                  10$:
508 015250  112737  000000  001543  MOVB    #RMCS1,PUTINX           ;SETUP PUT INDEX TABLE
015256  112737  000200  001544      MOVB    #200,PUTINX+1           ;SET TERMINATOR BYTE
015264  012737  000007  001410      MOV     #RECAL!GO,RMCS10        ;SET RMCS1 OUTPUT BUFFER = RECAL!GO
015272  004737  044424              JSR     PC,PUT                  ;GO WRITE RMCS1 VIA PUT SUBROUTINE
```

```
                015276  000404                          BR      11$             ;GO TO 11$ IF NO ERROR
                015300  000240                          NOP                     ;RETURN HERE IF ERROR
                015302  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                015304  000137  015404                  JMP     15$             ;GO TO 15$ IF ERROR
    509         015310  004737  044070          11$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
    510         015314  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR RECAL TO COMPLETE
    511
    512         015320  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                015324  000404                          BR      12$             ;GO TO 12$ IF NO ERROR
                015326  000240                          NOP                     ;RETURN HERE IF ERROR
                015330  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                015332  000137  015404                  JMP     15$             ;GO TO 15$ IF ERROR
    513         015336                          12$:
    514         015336  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                015342  000405                          BR      13$             ;GO TO 13$ IF NO ERROR
                015344  000240                          NOP                     ;RETURN HERE IF ERROR
                015346  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                015350  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                015352  000137  015404                  JMP     15$             ;GO TO 15$ IF ERROR
    515         015356  032737  010000  001376  13$:    BIT     #IVC,RMER2I     ;IVC SHOULD BE SET
    516         015364  001001                          BNE     14$             ;IT IS - GO CHECK SECONDARY ERRORS
    517         015366  104215                          EMT     215
    518         015370                          14$:
    519         015370  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                015374  000403                          BR      15$             ;GO TO 15$ IF NO ERROR
                015376  000240                          NOP                     ;RETURN HERE IF ERROR
                015400  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                015402  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    520         015404                          15$:
    521
    522                                         ;;*****************************************************************
                                                ;*TEST 16       IAE RECALIBRATE TEST
                                                ;;*****************************************************************
                015404                          TST16:
                015404  000004                          SCOPE                   ;SCOPE CALL
                015406  000240                          NOP                     ;START OF TEST
                015410  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                015414  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                015420  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                015424  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
                015432  012737  000016  001226          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    523
    524         015440  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                015444  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                015446  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
                015450  000240                          NOP                     ;RETURN HERE IF ERROR
                015452  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                015454  000137  015646                  JMP     120$            ;GO TO 120$ IF ERROR
    525         015460                          70$:
    526         015460  012737  177777  001444          MOV     #-1,RMDC0       ;RMDC WILL BE ALL ONES
    527         015466  012737  177777  001416          MOV     #-1,RMDA0       ;RMDA WILL BE ALL ONES
```

```
    528 015474  012737  000007  001410          MOV     #RECAL!GO,RMCS10
    529 015502  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO PUT INDEX TABLE
    530 015506  112722  000034                  MOVB    #RMDC,(R2)+     ;SETUP PUT INDEX TABLE
    531 015512  112722  000006                  MOVB    #RMDA,(R2)+
    532 015516  112722  000000                  MOVB    #RMCS1,(R2)+
    533 015522  112722  000200                  MOVB    #200,(R2)+
    534 015526  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        015532  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        015534  000240                          NOP                     ;RETURN HERE IF ERROR
        015536  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        015540  000137  015646                  JMP     120$            ;GO TO 120$ IF ERROR
    535 015544  004737  044070          80$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
    536 015550  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
    537
    538 015554  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        015560  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        015562  000240                          NOP                     ;RETURN HERE IF ERROR
        015564  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        015566  000137  015646                  JMP     120$            ;GO TO 120$ IF ERROR
    539 015572                          90$:
    540 015572  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        015576  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
        015600  000240                          NOP                     ;RETURN HERE IF ERROR
        015602  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        015604  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015606  000137  015646                  JMP     120$            ;GO TO 120$ IF ERROR
    541 015612                          100$:
    542 015612  004737  055322                  JSR     PC,RCLSTS       ;GO VERIFY RECALIBRATE OPERATION
        015616  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        015620  000240                          NOP                     ;RETURN HERE IF ERROR
        015622  104000                          EMT                     ;ERROR # DEFINED BY RCLSTS SUBROUTINE
        015624  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        015626  000137  015646                  JMP     120$            ;GO TO 120$ IF ERROR
    543 015632                          110$:
    544 015632  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        015636  000403                          BR      120$            ;GO TO 120$ IF NO ERROR
        015640  000240                          NOP                     ;RETURN HERE IF ERROR
        015642  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        015644  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    545 015646                          120$:
    546
    547                                 ;;*******************************************************************
                                        ;*TEST 17        RECALIBRATE AT OFFSET
                                        ;;*******************************************************************
        015646                          TST17:
        015646  000004                          SCOPE                   ;SCOPE CALL
        015650  000240                          NOP                     ;START OF TEST
        015652  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        015656  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        015662  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        015666  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
        015674  012737  000017  001226          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    548
    549 015702  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        015706  050020                          .WORD   050020          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
```

```
                                                        ;VERIFY PACK ACKNOWLEDGE
     015710  000404                             BR      5$              ;GO TO 5$ IF NO ERROR
     015712  000240                             NOP                     ;RETURN HERE IF ERROR
     015714  104000                             EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     015716  000137  016124                     JMP     80$             ;GO TO 80$ IF ERROR
550  015722                             5$:
551  015722  112737  000000  001543             MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
     015730  112737  000200  001544             MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     015736  012737  000015  001410             MOV     #OFFSET!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET.GO
     015744  004737  044424                     JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     015750  000404                             BR      10$             ;GO TO 10$ IF NO ERROR
     015752  000240                             NOP                     ;RETURN HERE IF ERROR
     015754  104000                             EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     015756  000137  016124                     JMP     80$             ;GO TO 80$ IF ERROR
552  015762                             10$:
553  015762  112737  000000  001543             MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
     015770  112737  000200  001544             MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     015776  012737  000007  001410             MOV     #RECAL!GO,RMCS10        ;SET RMCS1 OUTPUT BUFFER = RECAL.GO
     016004  004737  044424                     JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     016010  000404                             BR      20$             ;GO TO 20$ IF NO ERROR
     016012  000240                             NOP                     ;RETURN HERE IF ERROR
     016014  104000                             EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     016016  000137  016124                     JMP     80$             ;GO TO 80$ IF ERROR
554  016022                             20$:
555
556                                     ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     016022  004737  044070                     JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
557
558                                     ;WAIT FOR COMMAND TO COMPLETE
     016026  004737  044766                     JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
559
560  016032  004737  044154                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     016036  000404                             BR      50$             ;GO TO 50$ IF NO ERROR
     016040  000240                             NOP                     ;RETURN HERE IF ERROR
     016042  104000                             EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     016044  000137  016124                     JMP     80$             ;GO TO 80$ IF ERROR
561  016050                             50$:
562  016050  004737  045152                     JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     016054  000405                             BR      60$             ;GO TO 60$ IF NO ERROR
     016056  000240                             NOP                     ;RETURN HERE IF ERROR
     016060  104000                             EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     016062  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     016064  000137  016124                     JMP     80$             ;GO TO 80$ IF ERROR
563  016070                             60$:
564  016070  004737  055322                     JSR     PC,RCLSTS       ;GO VERIFY RECALIBRATE OPERATION
     016074  000405                             BR      70$             ;GO TO 70$ IF NO ERROR
     016076  000240                             NOP                     ;RETURN HERE IF ERROR
     016100  104000                             EMT                     ;ERROR # DEFINED BY RCLSTS SUBROUTINE
     016102  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     016104  000137  016124                     JMP     80$             ;GO TO 80$ IF ERROR
565  016110                             70$:
566  016110  004737  046004                     JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     016114  000403                             BR      80$             ;GO TO 80$ IF NO ERROR
     016116  000240                             NOP                     ;RETURN HERE IF ERROR
     016120  104000                             EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     016122  004736                             JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
567  016124                             80$:
```

```
568
569                                     ;;*****************************************************************
                                        ;*TEST 20        DRIVE CLEAR TEST
                                        ;;*****************************************************************
016124                                  TST20:
016124  000004                                SCOPE                   ;SCOPE CALL
016126  000240                                NOP                     ;START OF TEST
016130  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
016134  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
016140  013701  001464                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
016144  012737  000020  001226                MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
570
571 016152  004737  043150                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
016156  054130                              .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
016160  000404                                BR      2$              ;GO TO 2$ IF NO ERROR
016162  000240                                NOP                     ;RETURN HERE IF ERROR
016164  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
016166  000137  016374                        JMP     21$             ;GO TO 21$ IF ERROR
572 016172                              2$:
573 016172  012737  000000  001416            MOV     #0,RMDA0
574 016200  012737  000011  001410            MOV     #DRVCLR!GO,RMCS10
575 016206  012737  177777  001424            MOV     #-1,RMER10
576 016214  012737  177777  001452            MOV     #-1,RMER20
577 016222  042737  004000  001452            BIC     #LSC,RMER20     ;DELETE FOR PASS 2 ETCH
578 016230  012702  001543                    MOV     #PUTINX,R2      ;R2 POINTS TO INDEX TABLE
579 016234  112722  000006                    MOVB    #RMDA,(R2)+     ;RMDA CLEARS LBT
580 016240  112722  000042                    MOVB    #RMER2,(R2)+
581 016244  112722  000014                    MOVB    #RMER1,(R2)+
582 016250  112722  000000                    MOVB    #RMCS1,(R2)+
583 016254  112722  000200                    MOVB    #200,(R2)+
584 016260  004737  044424                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
016264  000403                                BR      3$              ;GO TO 3$ IF NO ERROR
016266  000240                                NOP                     ;RETURN HERE IF ERROR
016270  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
585 016272  000440                            BR      21$             ;ESCAPE IF ERROR
586 016274  004737  044070              3$:   JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
587 016300  004737  044766                    JSR     PC,TIMOUT       ;WAIT FOR DRIVE CLEAR TO COMPLETE
588
589 016304  004737  044154                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
016310  000403                                BR      4$              ;GO TO 4$ IF NO ERROR
016312  000240                                NOP                     ;RETURN HERE IF ERROR
016314  104000                                EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
590 016316  000426                            BR      21$             ;SKIP REMAINDER OF TEST
591 016320                              4$:
592 016320  004737  045152                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
016324  000405                                BR      5$              ;GO TO 5$ IF NO ERROR
016326  000240                                NOP                     ;RETURN HERE IF ERROR
016330  104000                                EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
016332  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016334  000137  016374                        JMP     21$             ;GO TO 21$ IF ERROR
593 016340                              5$:
```

```
                                                                N  8
 594 016340  004737  057064                 JSR     PC,DRVSTS       ;GO VERIFY DRIVE CLEAR
     016344  000405                         BR      6$              ;GO TO 6$ IF NO ERROR
     016346  000240                         NOP                     ;RETURN HERE IF ERROR
     016350  104000                         EMT                     ;ERROR # DEFINED BY DRVSTS SUBROUTINE
     016352  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     016354  000137  016374                 JMP     21$             ;GO TO 21$ IF ERROR
 595 016360                          6$:
 596 016360  004737  046004                 JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     016364  000403                         BR      21$             ;GO TO 21$ IF NO ERROR
     016366  000240                         NOP                     ;RETURN HERE IF ERROR
     016370  104000                         EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     016372  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
 597 016374                          21$:
 598
 599                                 ;;*******************************************************************
                                     ;*TEST 21       NOP TEST
                                     ;;*******************************************************************
     016374                          TST21:
     016374  000004                         SCOPE                   ;SCOPE CALL
     016376  000240                         NOP                     ;START OF TEST
     016400  012706  001100                 MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     016404  013700  001276                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     016410  013701  001464                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     016414  012737  000021  001226         MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 600
 601 016422  004737  043150                 JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     016426  054130                         .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
                                                                    ;VERIFY CONTROLLER CLEAR OPERATION
                                                                    ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                    ;VERIFY PACK ACKNOWLEDGE
                                                                    ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                    ;VERIFY RECALIBRATION
     016430  000404                         BR      70$             ;GO TO 70$ IF NO ERROR
     016432  000240                         NOP                     ;RETURN HERE IF ERROR
     016434  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     016436  000137  016624                 JMP     130$            ;GO TO 130$ IF ERROR
 602 016442                          70$:
 603 016442  112737  000000  001543         MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
     016450  112737  000200  001544         MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     016456  012737  000001  001410         MOV     #NOP!GO,RMCS10  ;SET RMCS1 OUTPUT BUFFER = NOP!GO
     016464  004737  044424                 JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     016470  000404                         BR      80$             ;GO TO 80$ IF NO ERROR
     016472  000240                         NOP                     ;RETURN HERE IF ERROR
     016474  104000                         EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     016476  000137  016624                 JMP     130$            ;GO TO 130$ IF ERROR
 604 016502  004737  044070          80$:   JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
 605
 606 016506  004737  044154                 JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     016512  000404                         BR      90$             ;GO TO 90$ IF NO ERROR
     016514  000240                         NOP                     ;RETURN HERE IF ERROR
     016516  104000                         EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     016520  000137  016624                 JMP     130$            ;GO TO 130$ IF ERROR
 607 016524                          90$:
 608 016524  004737  045152                 JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     016530  000405                         BR      100$            ;GO TO 100$ IF NO ERROR
     016532  000240                         NOP                     ;RETURN HERE IF ERROR
```

```
016534  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
016536  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016540  000137  016624                  JMP     130$            ;GO TO 130$ IF ERROR
609 016544                      100$:
610 016544  004737  051644              JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
016550  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
016552  000010                          .WORD   DPE             ;MASK FOR RMER2
016554  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
016556  000240                          NOP                     ;RETURN HERE IF ERROR
016560  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
016562  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016564  000137  016624                  JMP     130$            ;GO TO 130$ IF ERROR
611 016570                      110$:
612 016570  004737  061232              JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
016574  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
016576  000240                          NOP                     ;RETURN HERE IF ERROR
016600  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
016602  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
016604  000137  016624                  JMP     130$            ;GO TO 130$ IF ERROR
613 016610                      120$:
614 016610  004737  046004              JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
016614  000403                          BR      130$            ;GO TO 130$ IF NO ERROR
016616  000240                          NOP                     ;RETURN HERE IF ERROR
016620  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
016622  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
615 016624                      130$:
616
617                             ;;*******************************************************************
                                ;*TEST 22        OFFSET TEST
                                ;;*******************************************************************
016624                          TST22:
016624  000004                          SCOPE                   ;SCOPE CALL
016626  000240                          NOP                     ;START OF TEST
016630  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
016634  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
016640  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
016644  012737  000022  001226          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
618
619 016652  004737  043150              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
016656  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
016660  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
016662  000240                          NOP                     ;RETURN HERE IF ERROR
016664  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
016666  000137  017150                  JMP     100$            ;GO TO 100$ IF ERROR
620 016672                      40$:
621 016672  112737  000000  001543      MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
016700  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
016706  012737  000015  001410          MOV     #OFFSET!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET!GO
016714  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
016720  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
016722  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        016724  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        016726  000137  017150                  JMP     100$            ;GO TO 100$ IF ERROR
    622 016732  004737  044070          50$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
    623 016736  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
    624
    625 016742  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        016746  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
        016750  000240                          NOP                     ;RETURN HERE IF ERROR
        016752  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        016754  000137  017150                  JMP     100$            ;GO TO 100$ IF ERROR
    626 016760                          60$:
    627 016760  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        016764  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        016766  000240                          NOP                     ;RETURN HERE IF ERROR
        016770  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        016772  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        016774  000137  017150                  JMP     100$            ;GO TO 100$ IF ERROR
    628 017000  032737  000001  001346  70$:    BIT     #OM,RMDSI       ;OFFSET MODE ON??
    629 017006  001012                          BNE     80$             ;YES!!
    630 017010  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    631 017016  042737  177776  001142          BIC     #^COM,$BDDAT
    632 017024  012737  000001  001140          MOV     #OM,$GDDAT      ;GOOD DATA FOR TYPEOUT
    633 017032  104156                          EMT     156
    634 017034  032737  100000  001346  80$:    BIT     #ATA,RMDSI      ;WAS ATTENTION SET ??
    635 017042  001012                          BNE     90$             ;YES!.
    636 017044  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    637 017052  042737  077777  001142          BIC     #^CATA,$BDDAT
    638 017060  012737  100000  001140          MOV     #ATA,$GDDAT     ;GOOD DATA FOR TYPEOUT
    639 017066  104172                          EMT     172
    640 017070                          90$:
    641 017070  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        017074  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
        017076  000010                          .WORD   DPE             ;MASK FOR RMER2
        017100  000405                          BR      91$             ;GO TO 91$ IF NO ERROR
        017102  000240                          NOP                     ;RETURN HERE IF ERROR
        017104  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        017106  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017110  000137  017150                  JMP     100$            ;GO TO 100$ IF ERROR
    642 017114                          91$:
    643 017114  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        017120  000405                          BR      92$             ;GO TO 92$ IF NO ERROR
        017122  000240                          NOP                     ;RETURN HERE IF ERROR
        017124  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        017126  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017130  000137  017150                  JMP     100$            ;GO TO 100$ IF ERROR
    644 017134                          92$:
    645 017134  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        017140  000403                          BR      100$            ;GO TO 100$ IF NO ERROR
        017142  000240                          NOP                     ;RETURN HERE IF ERROR
        017144  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        017146  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    646 017150                          100$:
    647
    648                                 ;;*****************************************************************
                                        ;*TEST 23       GO/ATA TEST
                                        ;;*****************************************************************
        017150                          TST23:
```

```
                017150  000004                          SCOPE                   ;SCOPE CALL
                017152  000240                          NOP                     ;START OF TEST
                017154  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                017160  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                017164  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                017170  012737  000023  001226          MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
            649
            650 017176  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                017202  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                017204  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
                017206  000240                          NOP                     ;RETURN HERE IF ERROR
                017210  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                017212  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
            651 017216                          70$:
            652 017216  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
                017224  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                017232  012737  000015  001410          MOV     #OFFSET!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET!GO
                017240  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
                017244  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                017246  000240                          NOP                     ;RETURN HERE IF ERROR
                017250  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                017252  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
            653 017256  004737  044070          80$:    JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
            654
            655 017262  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                017266  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
                017270  000240                          NOP                     ;RETURN HERE IF ERROR
                017272  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                017274  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
            656 017300  032737  100000  001346  90$:    BIT     #ATA,RMDSI      ;IS ATTENTION SET??
            657 017306  001012                          BNE     100$            ;YES!!
            658 017310  012737  100000  001140          MOV     #ATA,$GDDAT     ;GOOD DATA FOR TYPEOUT
            659 017316  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
            660 017324  042737  077777  001142          BIC     #^CATA,$BDDAT
            661 017332  104172                          EMT     172
            662 017334                          100$:
            663 017334  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
                017340  000000                          .WORD   0               ;MASK FOR RMER1
                017342  000000                          .WORD   0               ;MASK FOR RMER2
                017344  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
                017346  000240                          NOP                     ;RETURN HERE IF ERROR
                017350  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
                017352  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                017354  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
            664 017360                          110$:
            665 017360  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
                017366  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                017374  012737  000001  001410          MOV     #NOP!GO,RMCS10  ;SET RMCS1 OUTPUT BUFFER = NOP!GO
                017402  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
                017406  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
                017410  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        017412  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        017414  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
    666 017420                          120$:
    667 017420  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        017424  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        017426  000240                          NOP                     ;RETURN HERE IF ERROR
        017430  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        017432  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
    668 017436                          130$:
    669 017436  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        017442  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        017444  000240                          NOP                     ;RETURN HERE IF ERROR
        017446  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        017450  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017452  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
    670 017456  032737  100000  001346  140$:  BIT     #ATA,RMDSI      ;IS ATTENTION RESET??
    671 017464  001411                          BEQ     150$            ;YES
    672 017466  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    673 017474  042737  077777  001142          BIC     #^CATA,$BDDAT
    674 017502  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
    675 017506  104246                          EMT     246
    676 017510                          150$:
    677 017510  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        017514  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
        017516  000010                          .WORD   DPE             ;MASK FOR RMER2
        017520  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        017522  000240                          NOP                     ;RETURN HERE IF ERROR
        017524  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        017526  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017530  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
    678 017534                          160$:
    679 017534  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        017540  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
        017542  000240                          NOP                     ;RETURN HERE IF ERROR
        017544  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        017546  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        017550  000137  017570                  JMP     180$            ;GO TO 180$ IF ERROR
    680 017554                          170$:
    681 017554  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        017560  000403                          BR      180$            ;GO TO 180$ IF NO ERROR
        017562  000240                          NOP                     ;RETURN HERE IF ERROR
        017564  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        017566  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    682 017570                          180$:
    683
    684                                 ;;*****************************************************************
                                        ;*TEST 24        WRITE ATA TEST
                                        ;;*****************************************************************
        017570                          TST24:
        017570  000004                          SCOPE                   ;SCOPE CALL
        017572  000240                          NOP                     ;START OF TEST
        017574  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        017600  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        017604  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        017610  012737  000024  001226          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    685
    686 017616  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
```

```
        017622  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        017624  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
        017626  000240                          NOP                     ;RETURN HERE IF ERROR
        017630  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        017632  000137  020134                  JMP     150$            ;GO TO 150$ IF ERROR
    687 017636                          70$:
    688 017636  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
        017644  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        017652  012737  000015  001410          MOV     #OFFSET.GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET.GO
        017660  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
        017664  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        017666  000240                          NOP                     ;RETURN HERE IF ERROR
        017670  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        017672  000137  020134                  JMP     150$            ;GO TO 150$ IF ERROR
    689 017676  004737  044070          80$:    JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
    690
    691 017702  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        017706  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        017710  000240                          NOP                     ;RETURN HERE IF ERROR
        017712  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        017714  000137  020134                  JMP     150$            ;GO TO 150$ IF ERROR
    692 017720  032737  100000  001346  90$:    BIT     #ATA,RMDSI      ;IS ATTENTION SET??
    693 017726  001012                          BNE     100$            ;YES!!
    694 017730  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    695 017736  042737  077777  001142          BIC     #^CATA,$BDDAT
    696 017744  012737  100000  001140          MOV     #ATA,$GDDAT     ;GOOD DATA FOR TYPEOUT
    697 017752  104172                          EMT     172
    698 017754  116102  000001          100$:   MOVB    1(R1),R2        ;R2 = ATTENTION BIT
    699 017760  042702  177400                  BIC     #^CATNMSK,R2    ;CLEAR SIGN EXTENSION
    700 017764  000240                          NOP
    701 017766  010237  001426                  MOV     R2,RMASO        ;PUT RMAS BIT IN OUTPUT BUF
    702 017772  112737  000016  001543          MOVB    #RMAS,PUTINX    ;WRITE REGISTER INDEX IN TABLE
    703 020000  112737  000200  001544          MOVB    #200,PUTINX+1   ;WRITE TERMINATOR
    704 020006  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        020012  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
        020014  000240                          NOP                     ;RETURN HERE IF ERROR
        020016  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        020020  000137  020134                  JMP     150$            ;GO TO 150$ IF ERROR
    705 020024                          110$:
    706 020024  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        020030  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
        020032  000240                          NOP                     ;RETURN HERE IF ERROR
        020034  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        020036  000137  020134                  JMP     150$            ;GO TO 150$ IF ERROR
    707 020042                          120$:
    708 020042  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        020046  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
        020050  000240                          NOP                     ;RETURN HERE IF ERROR
        020052  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        020054  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020056  000137  020134                  JMP     150$            ;GO TO 150$ IF ERROR
```

```
                                                ;;*****************************************************************
                                                ;*TEST 25       ERROR/ATA TEST
                                                ;;*****************************************************************
020134                                  TST25:
020134  000004                                    SCOPE                   ;SCOPE CALL
020136  000240                                    NOP                     ;START OF TEST
020140  012706  001100                            MOV     #STACK,SP       ;INITIALIZE STACK POINTER
020144  013700  001276                            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
020150  013701  001464                            MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
020154  012737  000025  001226                    MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
020162  004737  043150                            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
020166  054130                                    .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
020170  000404                                    BR      70$             ;GO TO 70$ IF NO ERROR
020172  000240                                    NOP                     ;RETURN HERE IF ERROR
020174  104000                                    EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
020176  000137  020526                            JMP     160$            ;GO TO 160$ IF ERROR
020202                                  70$:
020202  112737  000000  001543                    MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
020210  112737  000200  001544                    MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
020216  012737  000015  001410                    MOV     #OFFSET!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET!GO
020224  004737  044424                            JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
020230  000404                                    BR      80$             ;GO TO 80$ IF NO ERROR
020232  000240                                    NOP                     ;RETURN HERE IF ERROR
020234  104000                                    EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
020236  000137  020526                            JMP     160$            ;GO TO 160$ IF ERROR
020242  004737  044070                    80$:    JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS

020246  004737  044154                            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
020252  000404                                    BR      90$             ;GO TO 90$ IF NO ERROR
020254  000240                                    NOP                     ;RETURN HERE IF ERROR
020256  104000                                    EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
020260  000137  020526                            JMP     160$            ;GO TO 160$ IF ERROR
020264  032737  100000  001346            90$:    BIT     #ATA,RMDSI      ;IS ATA SET??
020272  001012                                    BNE     100$            ;YES!!
```

```
    729 020274  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    730 020302  042737  077777  001142          BIC     #^CATA,$BDDAT
    731 020310  012737  100000  001140          MOV     #ATA,$GDDAT     ;GOOD DATA FOR TYPEOUT
    732 020316  104172                          EMT     172
    733 020320                          100$:
    734 020320  112737  000014  001543          MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
        020326  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        020334  012737  040000  001424          MOV     #UNS,RMER10     ;SET RMER1 OUTPUT BUFFER = UNS
        020342  004737  044424                  JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
        020346  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
        020350  000240                          NOP                     ;RETURN HERE IF ERROR
        020352  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        020354  000137  020526                  JMP     160$            ;GO TO 160$ IF ERROR
    735 020360                          110$:
    736 020360  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
        020366  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        020374  012737  000001  001410          MOV     #NOP!GO,RMCS10  ;SET RMCS1 OUTPUT BUFFER = NOP!GO
        020402  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
        020406  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
        020410  000240                          NOP                     ;RETURN HERE IF ERROR
        020412  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        020414  000137  020526                  JMP     160$            ;GO TO 160$ IF ERROR
    737 020420                          120$:
    738 020420  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        020424  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        020426  000240                          NOP                     ;RETURN HERE IF ERROR
        020430  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        020432  000137  020526                  JMP     160$            ;GO TO 160$ IF ERROR
    739 020436  032737  100000  001346  130$:   BIT     #ATA,RMDSI      ;IS ATA STILL SET??
    740 020444  001012                          BNE     140$            ;YES!!
    741 020446  012737  100000  001140          MOV     #ATA,$GDDAT     ;GOOD DATA FOR TYPEOUT
    742 020454  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    743 020462  042737  077777  001142          BIC     #^CATA,$BDDAT
    744 020470  104250                          EMT     250
    745 020472                          140$:
    746 020472  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        020476  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        020500  000240                          NOP                     ;RETURN HERE IF ERROR
        020502  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        020504  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        020506  000137  020526                  JMP     160$            ;GO TO 160$ IF ERROR
    747 020512                          150$:
    748 020512  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        020516  000403                          BR      160$            ;GO TO 160$ IF NO ERROR
        020520  000240                          NOP                     ;RETURN HERE IF ERROR
        020522  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        020524  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    749 020526                          160$:
    750
    751                                 ;;*****************************************************************
                                        ;*TEST 26        PROGRAM INTERRUPT TEST
                                        ;;*****************************************************************
        020526                          TST26:
        020526  000004                          SCOPE                   ;SCOPE CALL
        020530  000240                          NOP                     ;START OF TEST
        020532  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        020536  013700  001276                  MOV     $BASE,R0        ;R0 - UNIBUS ADDRESS
```

```
                020542  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                020546  012737  000026  001226          MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     752
     753        020554  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                020560  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF ''SKI'' OR ''PIP'' IS SET
                                                                                ;VERIFY RECALIBRATION
                020562  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
                020564  000240                          NOP                     ;RETURN HERE IF ERROR
                020566  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                020570  000137  021216                  JMP     210$            ;GO TO 210$ IF ERROR
     754        020574                          70$:
     755        020574  113702  001272                  MOVB    $VECT1,R2       ;R2 = INTERRUPT ADDRESS
     756        020600  042702  177400                  BIC     #^C<377>,R2     ;CLEAR SIGN EXTENSION
     757        020604  011204                          MOV     (R2),R4         ;SAVE HANDLER ADDRESS
     758        020606  016205  000002                  MOV     2(R2),R5        ;SAVE HANDLER PRIORITY
     759        020612  012712  021056                  MOV     #150$,(R2)      ;WRITE HANDLER ADDRESS AND
     760        020616  012762  000300  000002          MOV     #PR6,2(R2)      ;PRIORITY FOR THIS TEST
     761        020624  113703  001273                  MOVB    $VECT1+1,R3     ;R3 = INTERRUPT LEVEL
     762        020630  042703  177400                  BIC     #^C<377>,R3     ;CLEAR SIGN EXTENSION
     763        020634  000303                          SWAB    R3
     764        020636  005303                          DEC     R3              ;DECREMENT INTERRUPT LEVEL
     765        020640  100001                          BPL     80$
     766        020642  005003                          CLR     R3
     767        020644  042703  177437          80$:    BIC     #^CPR7,R3
     768        020650  000240                          NOP
     769        020652  012746  000300                  MOV     #PR6,-(SP)      ;;PUSH #PR6 ON STACK
     770        020656  012746  021014                  MOV     #120$,-(SP)     ;;PUSH #120$ ON STACK
     771        020662  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
     772        020664  012746  021006                  MOV     #110$,-(SP)     ;;PUSH #110$ ON STACK
     773        020670  012737  177777  001426          MOV     #-1,RMASO       ;WRITE ONES IN RMAS TO
     774        020676  112737  000016  001543          MOVB    #RMAS,PUTINX    ;CLEAR ALL ATTENTIONS
     775        020704  112737  000200  001543          MOVB    #200,PUTINX
     776
     777        020712  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                020716  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
                020720  000240                          NOP                     ;RETURN HERE IF ERROR
                020722  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                020724  000137  021216                  JMP     210$            ;GO TO 210$ IF ERROR
     778        020730                          90$:
     779        020730  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
                020736  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                020744  012737  000115  001410          MOV     #OFFSET!GO!IE,RMCS10    ;SET RMCS1 OUTPUT BUFFER = OFFSET!GO!IE
                020752  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
                020756  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
                020760  000240                          NOP                     ;RETURN HERE IF ERROR
                020762  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                020764  000137  021216                  JMP     210$            ;GO TO 210$ IF ERROR
     780        020770                          100$:
     781        020770  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
     782        020774  004737  044070                  JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
     783        021000  012703  000205                  MOV     #205,R3         ;R3 = GROSS TIMEOUT
     784        021004  000002                          RTI                     ;DROP CP PRIORITY
```

```
    785 021006  005303                110$:   DEC     R3              ;TIMEOUT THE INTERRUPT
    786 021010  100376                        BPL     110$
    787
    788                               ;TIMEOUT BEFORE INTERRUPT
    789
    790 021012  000002                        RTI                     ;RAISE CP PRIORITY
    791 021014                        120$:
    792 021014  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        021020  000404                        BR      130$            ;GO TO 130$ IF NO ERROR
        021022  000240                        NOP                     ;RETURN HERE IF ERROR
        021024  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        021026  000137  021216                JMP     210$            ;GO TO 210$ IF ERROR
    793 021032                        130$:
    794 021032  004737  045152                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        021036  000405                        BR      140$            ;GO TO 140$ IF NO ERROR
        021040  000240                        NOP                     ;RETURN HERE IF ERROR
        021042  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        021044  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        021046  000137  021216                JMP     210$            ;GO TO 210$ IF ERROR
    795 021052                        140$:
        021052  104251                        EMT     251
    796 021054  000423                        BR      180$
    797
    798 021056  022626                150$:   CMP     (SP)+,(SP)+     ;ADJUST STACK
    799 021060  012716  021066                MOV     #160$,(SP)      ;CHANGE RETURN ADDRESS
    800 021064  000002                        RTI                     ;RAISE CP PRIORITY
    801 021066                        160$:
    802 021066  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        021072  000404                        BR      170$            ;GO TO 170$ IF NO ERROR
        021074  000240                        NOP                     ;RETURN HERE IF ERROR
        021076  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        021100  000137  021216                JMP     210$            ;GO TO 210$ IF ERROR
    803 021104                        170$:
    804 021104  004737  045152                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        021110  000405                        BR      180$            ;GO TO 180$ IF NO ERROR
        021112  000240                        NOP                     ;RETURN HERE IF ERROR
        021114  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        021116  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        021120  000137  021216                JMP     210$            ;GO TO 210$ IF ERROR
    805 021124                        180$:
    806 021124  004737  051644                JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        021130  115760                        .WORD   NDTMSK          ;MASK FOR RMER1
        021132  000010                        .WORD   DPE             ;MASK FOR RMER2
        021134  000405                        BR      190$            ;GO TO 190$ IF NO ERROR
        021136  000240                        NOP                     ;RETURN HERE IF ERROR
        021140  104000                        EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        021142  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        021144  000137  021216                JMP     210$            ;GO TO 210$ IF ERROR
    807 021150                        190$:
    808 021150  004737  061232                JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        021154  000405                        BR      200$            ;GO TO 200$ IF NO ERROR
        021156  000240                        NOP                     ;RETURN HERE IF ERROR
        021160  104000                        EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        021162  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        021164  000137  021216                JMP     210$            ;GO TO 210$ IF ERROR
    809 021170                        200$:
    810 021170  004737  046004                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
```

```
021174  000405                          BR      205$            ;GO TO 205$ IF NO ERROR
021176  000240                          NOP                     ;RETURN HERE IF ERROR
021200  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
021202  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
021204  000137  021216                  JMP     210$            ;GO TO 210$ IF ERROR
811 021210                      205$:
812 021210  010412                      MOV     R4,(R2)
813 021212  010562  000002              MOV     R5,2(R2)
814 021216                      210$:
815
816                                     ;;*****************************************************************
                                        ;*TEST 27        INHIBIT INTERRUPT TEST
                                        ;;*****************************************************************
021216                          TST27:
021216  000004                          SCOPE                   ;SCOPE CALL
021220  000240                          NOP                     ;START OF TEST
021222  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
021226  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
021232  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
021236  012737  000027  001226          MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
817
818 021244  004737  043150              JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
021250  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
021252  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
021254  000240                          NOP                     ;RETURN HERE IF ERROR
021256  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
021260  000137  021564                  JMP     210$            ;GO TO 210$ IF ERROR
819 021264                      70$:
820 021264  113702  001272              MOVB    $VECT1,R2       ;R2 = INTERRUPT ADDRESS
821 021270  042702  177400              BIC     #^C<377>,R2     ;CLEAR SIGN EXTENSION
822 021274  011204                      MOV     (R2),R4         ;SAVE HANDLER ADDRESS
823 021276  016205  000002              MOV     2(R2),R5        ;SAVE HANDLER PRIORITY
824 021302  012712  021524              MOV     #130$,(R2)      ;WRITE HANDLER ADDRESS AND
825 021306  012762  000300  000002      MOV     #PR6,2(R2)      ;PRIORITY FOR THIS TEST
826 021314  113703  001273              MOVB    $VECT1+1,R3     ;R3 = INTERRUPT LEVEL
827 021320  042703  177400              BIC     #^C<377>,R3     ;CLEAR SIGN EXTENSION
828 021324  012746  000300              MOV     #PR6,-(SP)      ;;PUSH #PR6 ON STACK
829 021330  012746  021556              MOV     #180$,-(SP)     ;;PUSH #180$ ON STACK
830 021334  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
831 021336  005316                      DEC     (SP)
832 021340  100001                      BPL     80$
833 021342  005016                      CLR     (SP)
834 021344  042716  177437      80$:    BIC     #^CPR7,(SP)
835 021350  012746  021512              MOV     #120$,-(SP)     ;;PUSH #120$ ON STACK
836 021354  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
837 021356  012746  021440              MOV     #100$,-(SP)     ;;PUSH #100$ ON STACK
838 021362  004737  044070              JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
839
840 021366  112737  000000  001543      MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
021374  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
021402  012737  000115  001410          MOV     #OFFSET!GO!IE,RMCS10    ;SET RMCS1 OUTPUT BUFFER  OFFSET!GO!IE
```

```
021410  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
021414  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
021416  000240                          NOP                     ;RETURN HERE IF ERROR
021420  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
021422  000137  021564                  JMP     210$            ;GO TO 210$ IF ERROR
841 021426                      90$:
842 021426  004737  044766              JSR     PC,TIMOUT       ;WAIT FOR GO = 0
843 021432  012703  000205              MOV     #205,R3         ;R3 = GROSS TIMER
844 021436  000002                      RTI
845
846 021440  005303              100$:   DEC     R3              ;DELAY
847 021442  100376                      BPL     100$
848
849 021444  112737  000000  001543      MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
021452  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
021460  012737  000000  001410          MOV     #NOP,RMCS10     ;SET RMCS1 OUTPUT BUFFER = NOP
021466  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
021472  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
021474  000240                          NOP                     ;RETURN HERE IF ERROR
021476  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
021500  000137  021564                  JMP     210$            ;GO TO 210$ IF ERROR
850 021504  012703  000205      110$:   MOV     #205,R3
851 021510  000002                      RTI
852
853 021512  012712  021526      120$:   MOV     #140$,(R2)
854 021516  005303              125$:   DEC     R3
855 021520  100376                      BPL     125$
856 021522  000002                      RTI                     ;NO ERROR
857
858 021524  022626              130$:   CMP     (SP)+,(SP)+     ;ADJUST STACK
859 021526  022626              140$:   CMP     (SP)+,(SP)+     ;ADJUST STACK
860 021530  012716  021536              MOV     #160$,(SP)
861 021534  000002                      RTI
862 021536                      160$:
863 021536  004737  044154              JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
021542  000404                          BR      170$            ;GO TO 170$ IF NO ERROR
021544  000240                          NOP                     ;RETURN HERE IF ERROR
021546  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
021550  000137  021564                  JMP     210$            ;GO TO 210$ IF ERROR
864 021554                      170$:
021554  104252                          EMT     252
865
866 021556  010412              180$:   MOV     R4,(R2)
867 021560  010562  000002              MOV     R5,2(R2)
868 021564                      210$:
869
870                             ;;*****************************************************************
                                ;*TEST 30        RETURN TO CENTERLINE TEST
                                ;;*****************************************************************
021564                          TST30:
021564  000004                          SCOPE                   ;SCOPE CALL
021566  000240                          NOP                     ;START OF TEST
021570  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
021574  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
021600  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
021604  012737  000030  001226          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
871
```

```
    872 021612  004737  043150                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        021616  054130                        .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        021620  000404                        BR      70$             ;GO TO 70$ IF NO ERROR
        021622  000240                        NOP                     ;RETURN HERE IF ERROR
        021624  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        021626  000137  022106                JMP     150$            ;GO TO 150$ IF ERROR
    873 021632                          70$:
    874 021632  112737  000000  001543        MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
        021640  112737  000200  001544        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        021646  012737  000017  001410        MOV     #RTC!GO,RMCS10  ;SET RMCS1 OUTPUT BUFFER = RTC.GO
        021654  004737  044424                JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
        021660  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
        021662  000240                        NOP                     ;RETURN HERE IF ERROR
        021664  104000                        EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        021666  000137  022106                JMP     150$            ;GO TO 150$ IF ERROR
    875 021672  004737  044070          80$:  JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
    876 021676  004737  044766                JSR     PC,TIMOUT       ;WAIT FOR RECAL TO COMPLETE
    877
    878 021702  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        021706  000404                        BR      90$             ;GO TO 90$ IF NO ERROR
        021710  000240                        NOP                     ;RETURN HERE IF ERROR
        021712  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        021714  000137  022106                JMP     150$            ;GO TO 150$ IF ERROR
    879 021720                          90$:
    880 021720  004737  045152                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        021724  000405                        BR      100$            ;GO TO 100$ IF NO ERROR
        021726  000240                        NOP                     ;RETURN HERE IF ERROR
        021730  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        021732  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        021734  000137  022106                JMP     150$            ;GO TO 150$ IF ERROR
    881 021740  032737  000001  001346  100$: BIT     #OM,RMDSI       ;OFFSET MODE OFF??
    882 021746  001411                        BEQ     110$            ;YES!!
    883 021750  005037  001140                CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
    884 021754  013737  001346  001142        MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    885 021762  042737  177776  001142        BIC     #^COM,$BDDAT
    886 021770  104157                        EMT     157
    887 021772  032737  100000  001346  110$: BIT     #ATA,RMDSI      ;WAS ATTENTION SET ??
    888 022000  001012                        BNE     120$            ;YES!!
    889 022002  013737  001346  001142        MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    890 022010  042737  077777  001142        BIC     #^CATA,$BDDAT
    891 022016  012737  100000  001140        MOV     #ATA,$GDDAT     ;GOOD DATA FOR TYPEOUT
    892 022024  104171                        EMT     171
    893 022026                          120$:
    894 022026  004737  051644                JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        022032  115760                        .WORD   NDTMSK          ;MASK FOR RMER1
        022034  000010                        .WORD   DPE             ;MASK FOR RMER2
        022036  000405                        BR      130$            ;GO TO 130$ IF NO ERROR
        022040  000240                        NOP                     ;RETURN HERE IF ERROR
        022042  104000                        EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        022044  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022046  000137  022106                JMP     150$            ;GO TO 150$ IF ERROR
```

```
                                130$:
022052  004737  061232                 JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
022056  000405                         BR      140$            ;GO TO 140$ IF NO ERROR
022060  000240                         NOP                     ;RETURN HERE IF ERROR
022062  104000                         EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
022064  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
022066  000137  022106                 JMP     150$            ;GO TO 150$ IF ERROR
022072                          140$:
022072  004737  046004                 JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
022076  000403                         BR      150$            ;GO TO 150$ IF NO ERROR
022100  000240                         NOP                     ;RETURN HERE IF ERROR
022102  104000                         EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
022104  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
022106                          150$:

                                ;;*******************************************************************
                                ;*TEST 31       READ IN PRESET TEST
                                ;;*******************************************************************
022106                          TST31:
022106  000004                         SCOPE                   ;SCOPE CALL
022110  000240                         NOP                     ;START OF TEST
022112  012706  001100                 MOV     #STACK,SP       ;INITIALIZE STACK POINTER
022116  013700  001276                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
022122  013701  001464                 MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
022126  012737  000031  001226         MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

022134  004737  043150                 JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
022140  054130                         .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                               ;VERIFY PACK ACKNOWLEDGE
                                                               ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                               ;VERIFY RECALIBRATION
022142  000404                         BR      70$             ;GO TO 70$ IF NO ERROR
022144  000240                         NOP                     ;RETURN HERE IF ERROR
022146  104000                         EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
022150  000137  022470                 JMP     160$            ;GO TO 160$ IF ERROR
022154                          70$:
022154  012702  001543                 MOV     #PUTINX,R2      ;R2 POINTS TO INDEX TABLE
022160  112722  000006                 MOVB    #RMDA,(R2)+
022164  112722  000034                 MOVB    #RMDC,(R2)+
022170  112722  000032                 MOVB    #RMOF,(R2)+
022174  112722  000000                 MOVB    #RMCS1,(R2)+
022200  112722  000200                 MOVB    #200,(R2)+
022204  012737  177777  001416         MOV     #-1,RMDA0
022212  012737  177777  001444         MOV     #-1,RMDC0
022220  012737  016200  001442         MOV     #FMT16!ECI!HCI!OFD,RMOF0
022226  012737  000021  001410         MOV     #RIP!GO,RMCS10
022234  004737  044424                 JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
022240  000404                         BR      80$             ;GO TO 80$ IF NO ERROR
022242  000240                         NOP                     ;RETURN HERE IF ERROR
022244  104000                         EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
022246  000137  022470                 JMP     160$            ;GO TO 160$ IF ERROR
022252  004737  044070         80$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
022256  004737  044766                 JSR     PC,TIMOUT       ;WAIT FOR RIP TO COMPLETE
```

```
    919 022262  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        022266  000404                        BR      90$             ;GO TO 90$ IF NO ERROR
        022270  000240                        NOP                     ;RETURN HERE IF ERROR
        022272  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        022274  000137  022470                JMP     160$            ;GO TO 160$ IF ERROR
    920 022300                          90$:
    921 022300  004737  045152                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        022304  000405                        BR      100$            ;GO TO 100$ IF NO ERROR
        022306  000240                        NOP                     ;RETURN HERE IF ERROR
        022310  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        022312  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022314  000137  022470                JMP     160$            ;GO TO 160$ IF ERROR
    922 022320  013737  001366  001142  100$: MOV     RMOFI,$BDDAT    ;IS RMOF RESET??
    923 022326  042737  161577  001142        BIC     #^C<FMT16 ECI!HCI!OFD>,$BDDAT
    924 022334  001403                        BEQ     110$            ;YES!!
    925 022336  005037  001140                CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
    926 022342  104160                        EMT     160
    927 022344  005737  001342          110$: TST     RMDAI           ;IS RMDA RESET??
    928 022350  001406                        BEQ     120$            ;YES!!
    929 022352  013737  001342  001142        MOV     RMDAI,$BDDAT    ;BAD DATA FOR TYPEOUT
    930 022360  005037  001140                CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
    931 022364  104161                        EMT     161
    932 022366  005737  001370          120$: TST     RMDCI           ;IS RMDC RESET??
    933 022372  001406                        BEQ     130$            ;YES!!
    934 022374  013737  001370  001142        MOV     RMDCI,$BDDAT    ;BAD DATA FOR TYPEOUT
    935 022402  005037  001140                CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
    936 022406  104162                        EMT     162
    937 022410                          130$:
    938 022410  004737  051644                JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        022414  115760                        .WORD   NDTMSK          ;MASK FOR RMER1
        022416  000010                        .WORD   DPE             ;MASK FOR RMER2
        022420  000405                        BR      140$            ;GO TO 140$ IF NO ERROR
        022422  000240                        NOP                     ;RETURN HERE IF ERROR
        022424  104000                        EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        022426  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022430  000137  022470                JMP     160$            ;GO TO 160$ IF ERROR
    939 022434                          140$:
    940 022434  004737  061232                JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        022440  000405                        BR      150$            ;GO TO 150$ IF NO ERROR
        022442  000240                        NOP                     ;RETURN HERE IF ERROR
        022444  104000                        EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        022446  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        022450  000137  022470                JMP     160$            ;GO TO 160$ IF ERROR
    941 022454                          150$:
    942 022454  004737  046004                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        022460  000403                        BR      160$            ;GO TO 160$ IF NO ERROR
        022462  000240                        NOP                     ;RETURN HERE IF ERROR
        022464  104000                        EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        022466  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
    943 022470                          160$:
    944
    945                                 ;;*****************************************************************
                                        ;*TEST 32        RMDC CLEAR OFFSET TEST
                                        ;;*****************************************************************
        022470                          TST32:
        022470  000004                          SCOPE                   ;SCOPE CALL
        022472  000240                          NOP                     ;START OF TEST
```

```
                022474  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                022500  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                022504  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                022510  012737  000032  001226          MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
    946
    947         022516  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                022522  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                022524  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
                022526  000240                          NOP                     ;RETURN HERE IF ERROR
                022530  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                022532  000137  023006                  JMP     140$            ;GO TO 140$ IF ERROR
    948         022536                          70$:
    949         022536  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
                022544  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                022552  012737  000015  001410          MOV     #OFFSET.GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET!GO
                022560  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
                022564  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                022566  000240                          NOP                     ;RETURN HERE IF ERROR
                022570  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                022572  000137  023006                  JMP     140$            ;GO TO 140$ IF ERROR
    950         022576  004737  044070          80$:    JSR     PC,GETSTS       ;SETUP FOR STATUS
    951
    952         022602  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                022606  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
                022610  000240                          NOP                     ;RETURN HERE IF ERROR
                022612  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                022614  000137  023006                  JMP     140$            ;GO TO 140$ IF ERROR
    953         022620                          90$:
    954         022620  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                022624  000405                          BR      100$            ;GO TO 100$ IF NO ERROR
                022626  000240                          NOP                     ;RETURN HERE IF ERROR
                022630  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                022632  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                022634  000137  023006                  JMP     140$            ;GO TO 140$ IF ERROR
    955         022640  032737  000001  001346  100$:   BIT     #OM,RMDSI       ;OFFSET ON??
    956         022646  001013                          BNE     110$            ;YES
    957         022650  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
    958         022656  042737  177776  001142          BIC     #^COM,$BDDAT
    959         022664  012737  000001  001140          MOV     #OM,$GDDAT      ;GOOD DATA FOR TYPEOUT
    960         022672  104156                          EMT     156
    961         022674  000444                          BR      140$            ;SKIP REST OF TEST
    962         022676                          110$:
    963         022676  112737  000034  001543          MOVB    #RMDC,PUTINX    ;SETUP PUT INDEX TABLE
                022704  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
                022712  012737  000000  001444          MOV     #0,RMDC0        ;SET RMDC OUTPUT BUFFER = 0
                022720  004737  044424                  JSR     PC,PUT          ;GO WRITE RMDC VIA PUT SUBROUTINE
                022724  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
                022726  000240                          NOP                     ;RETURN HERE IF ERROR
                022730  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
                022732  000137  023006                  JMP     140$            ;GO TO 140$ IF ERROR
    964         022736                          120$:
```

```
965 022736  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
    022742  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
    022744  000240                          NOP                     ;RETURN HERE IF ERROR
    022746  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
    022750  000137  023006                  JMP     140$            ;GO TO 140$ IF ERROR
966 022754  032737  000001  001346  130$:   BIT     #OM,RMDSI       ;DID OFFSET MODE RESET??
967 022762  001411                          BEQ     140$
968 022764  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
969 022772  042737  177776  001142          BIC     #^COM,$BDDAT
970 023000  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
971 023004  104253                          EMT     253
972 023006                          140$:
973
974                                 ;;*****************************************************************
                                    ;*TEST 33       ILLEGAL FUNCTION TEST
                                    ;;*****************************************************************
    023006                          TST33:
    023006  000004                          SCOPE                   ;SCOPE CALL
    023010  000240                          NOP                     ;START OF TEST
    023012  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
    023016  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
    023022  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
    023026  012737  000033  001226          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
975
976 023034  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
    023040  040000                          .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                    ;CLEAR CONTROLLER & SELECT DEVICE
    023042  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
    023044  000240                          NOP                     ;RETURN HERE IF ERROR
    023046  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
    023050  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
977 023054                          5$:
978 023054  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS
979 023060  012702  000000                  MOV     #NOP,R2
980 023064                          10$:
981 023064  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
    023070  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
    023072  000240                          NOP                     ;RETURN HERE IF ERROR
    023074  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
    023076  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
982 023102                          20$:
983
984                                 ;SET DIAGNOSTIC MODE AND DIAGNOSTIC UNIT READY, DIAGNOSTIC ON
985                                 ;CYLINDER
986 023102  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
    023110  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
    023116  012737  000001  001434          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
    023124  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
    023130  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
    023132  000240                          NOP                     ;RETURN HERE IF ERROR
    023134  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
    023136  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
987 023142                          30$:
988 023142  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
    023150  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
    023156  012737  001401  001434          MOV     #DMD!MUR!MOC,RMMR10     ;SET RMMR1 OUTPUT BUFFER   DMD!MUR!MOC
    023164  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
```

```
        023170  000404                          BR      35$             ;GO TO 35$ IF NO ERROR
        023172  000240                          NOP                     ;RETURN HERE IF ERROR
        023174  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        023176  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
   989  023202                          35$:
   990
   991                                  ;SET VOLUME VALID BEFORE ISSUING FUNCTION CODE BEING TESTED
   992  023202  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
        023210  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        023216  012737  000023  001410          MOV     #PAKACK!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = PAKACK!GO
        023224  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
        023230  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        023232  000240                          NOP                     ;RETURN HERE IF ERROR
        023234  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        023236  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
   993  023242                          40$:
   994  023242  012737  000001  001410          MOV     #GO,RMCS10
   995  023250  050237  001410                  BIS     R2,RMCS10       ;WRITE FUNCTION CODE IN BUFFER
   996  023254  012737  103640  001414          MOV     #BUFONE,RMBA0   ;DUMMY BUS ADDRESS
   997  023262  012737  177777  001412          MOV     #-1,RMWC0       ;DUMMY WORD COUNT
   998  023270  005037  001416                  CLR     RMDA0           ;CLEAR DISK ADDRESS
   999  023274  005037  001444                  CLR     RMDC0           ;CLEAR CYLINDER ADDRESS
  1000  023300  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
  1001  023306  012703  001543                  MOV     #PUTINX,R3      ;WRITE REGISTER INDEX TABLE
  1002  023312  112723  000004                  MOVB    #RMBA,(R3)+
  1003  023316  112723  000002                  MOVB    #RMWC,(R3)+
  1004  023322  112723  000032                  MOVB    #RMOF,(R3)+
  1005  023326  112723  000006                  MOVB    #RMDA,(R3)+
  1006  023332  112723  000034                  MOVB    #RMDC,(R3)+
  1007  023336  112723  000000                  MOVB    #RMCS1,(R3)+
  1008  023342  112713  000200                  MOVB    #200,(R3)
  1009  023346  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        023352  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        023354  000240                          NOP                     ;RETURN HERE IF ERROR
        023356  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        023360  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
  1010  023364  004737  044766          50$:    JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
  1011
  1012  023370  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        023374  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
        023376  000240                          NOP                     ;RETURN HERE IF ERROR
        023400  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        023402  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
  1013  023406                          60$:
  1014  023406  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        023412  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        023414  000240                          NOP                     ;RETURN HERE IF ERROR
        023416  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        023420  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        023422  000137  023572                  JMP     110$            ;GO TO 110$ IF ERROR
  1015  023426  016237  067046  001140  70$:    MOV     FNCDTB(R2),$GDDAT       ;GET ENTRY FROM FUNCTION CODE TABLE
  1016  023434  042737  177776  001140          BIC     #^CILF,$GDDAT
  1017  023442  013737  001350  001142          MOV     RMER11,$BDDAT   ;BAD DATA FOR TYPEOUT
  1018  023450  042737  177776  001142          BIC     #^CILF,$BDDAT
  1019  023456  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ILF STATUS CORRECT?
  1020  023464  001402                          BEQ     80$             ;YES!!
  1021  023466  104254                          EMT     254
```

```
 1022 023470  000440                          BR      110$
 1023 023472  016237  067046  001140  80$:    MOV     FNCDTB(R2),$GDDAT
 1024 023500  032737  040000  001346          BIT     #ERR,RMDSI      ;WAS AN ERROR DETECTED??
 1025 023506  001403                          BEQ     90$             ;NO!!
 1026 023510  052737  100000  001140          BIS     #ATA,$GDDAT     ;YES - ATA SHOULD BE ON
 1027 023516  042737  077777  001140  90$:    BIC     #^CATA,$GDDAT
 1028 023524  013737  001346  001142          MOV     RMDSI,$BDDAT    ;GET DRIVE'S ATTENTION
 1029 023532  042737  077777  001142          BIC     #^CATA,$BDDAT
 1030 023540  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA STATUS OK??
 1031 023546  001402                          BEQ     100$            ;YES!!
 1032 023550  104255                          EMT     255
 1033 023552  000407                          BR      110$
 1034 023554  062702  000002          100$:   ADD     #2,R2           ;GO TO NEXT FUNCTION CODE
 1035 023560  022702  000076                  CMP     #ILF76,R2       ;DONE??
 1036 023564  103402                          BLO     110$            ;YES !!
 1037 023566  000137  023064                  JMP     10$             ;TEST NEXT FUNCTION
 1038 023572                          110$:
 1039
 1040                                 ;;*****************************************************************
                                      ;*TEST 34       INVALID COMMAND TEST
                                      ;;*****************************************************************
      023572                          TST34:
      023572  000004                          SCOPE                   ;SCOPE CALL
      023574  000240                          NOP                     ;START OF TEST
      023576  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
      023602  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
      023606  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
      023612  012737  000034  001226          MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1041
 1042 023620  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
      023624  040000                          .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                              ;CLEAR CONTROLLER & SELECT DEVICE
      023626  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
      023630  000240                          NOP                     ;RETURN HERE IF ERROR
      023632  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
      023634  000137  024316                  JMP     110$            ;GO TO 110$ IF ERROR
 1043 023640                          5$:
 1044 023640  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS
 1045 023644  012702  000000                  MOV     #NOP,R2
 1046 023650                          10$:
 1047 023650  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
      023654  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
      023656  000240                          NOP                     ;RETURN HERE IF ERROR
      023660  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
      023662  000137  024316                  JMP     110$            ;GO TO 110$ IF ERROR
 1048 023666                          20$:
 1049
 1050
 1051                                 ;SET DIAGNOSTIC MODE AND DIAGNOSTIC UNIT READY, DIAGNOSTIC ON
                                      ;CYLINDER
 1052 023666  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
      023674  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      023702  012737  000001  001434          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER - DMD
      023710  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
      023714  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
      023716  000240                          NOP                     ;RETURN HERE IF ERROR
      023720  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      023722  000137  024316                  JMP     110$            ;GO TO 110$ IF ERROR
```

```
1053 023726                                30$:
1054 023726  112737  000024  001543              MOVB    #RMMR1,PUTINX    ;SETUP PUT INDEX TABLE
     023734  112737  000200  001544              MOVB    #200,PUTINX+1    ;SET TERMINATOR BYTE
     023742  012737  001401  001434              MOV     #DMD!MUR!MOC,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD.MUR.MOC
     023750  004737  044424                      JSR     PC,PUT           ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     023754  000404                              BR      35$              ;GO TO 35$ IF NO ERROR
     023756  000240                              NOP                      ;RETURN HERE IF ERROR
     023760  104000                              EMT                      ;ERROR DEFINED BY PUT SUBROUTINE
     023762  000137  024316                      JMP     110$             ;GO TO 110$ IF ERROR
1055 023766                                35$:
1056
1057
1058                                       ;VOLUME VALID IS LEFT RESET FOR THE TEST
1059
1060 023766                                40$:
1061 023766  012737  000001  001410              MOV     #GO,RMCS10
1062 023774  050237  001410                      BIS     R2,RMCS10        ;WRITE FUNCTION CODE IN BUFFER
1063 024000  012737  103640  001414              MOV     #BUFONE,RMBA0    ;DUMMY BUS ADDRESS
1064 024006  012737  177777  001412              MOV     #-1,RMWC0        ;DUMMY WORD COUNT
1065 024014  005037  001416                      CLR     RMDA0            ;CLEAR DISK ADDRESS
1066 024020  005037  001444                      CLR     RMDC0            ;CLEAR CYLINDER ADDRESS
1067 024024  012737  010000  001442              MOV     #FMT16,RMOF0     ;16 BIT FORMAT
1068 024032  012703  001543                      MOV     #PUTINX,R3       ;WRITE REGISTER INDEX TABLE
1069 024036  112723  000004                      MOVB    #RMBA,(R3)+
1070 024042  112723  000002                      MOVB    #RMWC,(R3)+
1071 024046  112723  000032                      MOVB    #RMOF,(R3)+
1072 024052  112723  000006                      MOVB    #RMDA,(R3)+
1073 024056  112723  000034                      MOVB    #RMDC,(R3)+
1074 024062  112723  000000                      MOVB    #RMCS1,(R3)+
1075 024066  112713  000200                      MOVB    #200,(R3)
1076 024072  004737  044424                      JSR     PC,PUT           ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     024076  000404                              BR      50$              ;GO TO 50$ IF NO ERROR
     024100  000240                              NOP                      ;RETURN HERE IF ERROR
     024102  104000                              EMT                      ;ERROR # DEFINED BY PUT SUBROUTINE
     024104  000137  024316                      JMP     110$             ;GO TO 110$ IF ERROR
1077 024110  004737  044766          50$:        JSR     PC,TIMOUT        ;WAIT FOR GO TO RESET
1078
1079 024114  004737  044154                      JSR     PC,GET           ;GO READ REGISTER(S) WITH GET SUBROUTINE
     024120  000404                              BR      60$              ;GO TO 60$ IF NO ERROR
     024122  000240                              NOP                      ;RETURN HERE IF ERROR
     024124  104000                              EMT                      ;ERROR # DEFINED BY GET SUBROUTINE
     024126  000137  024316                      JMP     110$             ;GO TO 110$ IF ERROR
1080 024132                                60$:
1081 024132  004737  045152                      JSR     PC,PRIERR        ;GO CHECK FOR PRIMARY ERRORS
     024136  000405                              BR      70$              ;GO TO 70$ IF NO ERROR
     024140  000240                              NOP                      ;RETURN HERE IF ERROR
     024142  104000                              EMT                      ;ERROR # DEFINED BY PRIERR SUBROUTINE
     024144  004736                              JSR     PC,@(SP)+        ;GO BACK FOR MORE ERROR CHECKS
     024146  000137  024316                      JMP     110$             ;GO TO 110$ IF ERROR
1082 024152  016237  067046  001140  70$:        MOV     FNCDTB(R2),$GDDAT        ;GET ENTRY FROM FUNCTION CODE TABLE
1083 024160  042737  167777  001140              BIC     #^CIVC,$GDDAT
1084 024166  013737  001376  001142              MOV     RMER2I,$BDDAT    ;BAD DATA FOR TYPEOUT
1085 024174  042737  167777  001142              BIC     #^CIVC,$BDDAT
1086 024202  023737  001140  001142              CMP     $GDDAT,$BDDAT    ;IS IVC STATUS CORRECT?
1087 024210  001402                              BEQ     80$              ;YES.!
1088 024212  104216                              EMT     216
1089 024214  000440                              BR      110$
```

```
1090 024216 016237 067046 001140 80$:   MOV     FNCDTB(R2),$GDDAT
1091 024224 032737 040000 001346        BIT     #ERR,RMDSI      ;WAS AN ERROR DETECTED??
1092 024232 001403                      BEQ     90$             ;NO!!
1093 024234 052737 100000 001140        BIS     #ATA,$GDDAT     ;YES - ATA SHOULD BE ON
1094 024242 042737 077777 001140 90$:   BIC     #^CATA,$GDDAT
1095 024250 013737 001346 001142        MOV     RMDSI,$BDDAT    ;GET DRIVE'S ATTENTION
1096 024256 042737 077777 001142        BIC     #^CATA,$BDDAT
1097 024264 023737 001140 001142        CMP     $GDDAT,$BDDAT   ;IS ATA STATUS OK??
1098 024272 001402                      BEQ     100$            ;YES!!
1099 024274 104255                      EMT     255
1100 024276 000407                      BR      110$
1101 024300 062702 000002       100$:   ADD     #2,R2           ;GO TO NEXT FUNCTION CODE
1102 024304 022702 000076               CMP     #ILF76,R2       ;DONE??
1103 024310 103402                      BLO     110$            ;YES .!
1104 024312 000137 023650               JMP     10$             ;TEST NEXT FUNCTION
1105 024316                     110$:
1106
1107                            ;;*****************************************************************
                                ;*TEST 35        INVALID ADDRESS ERROR TEST
                                ;;*****************************************************************
     024316                     TST35:
     024316 000004                      SCOPE                   ;SCOPE CALL
     024320 000240                      NOP                     ;START OF TEST
     024322 012706 001100               MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     024326 013700 001276               MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     024332 013701 001464               MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     024336 012737 000035 001226        MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1108
1109 024344 004737 043150               JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     024350 040000                      .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
     024352 000404                      BR      5$              ;GO TO 5$ IF NO ERROR
     024354 000240                      NOP                     ;RETURN HERE IF ERROR
     024356 104000                      EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     024360 000137 025106               JMP     110$            ;GO TO 110$ IF ERROR
1110 024364                     5$:
1111 024364 004737 044070               JSR     PC,GETSTS       ;SETUP FOR STATUS
1112 024370 012702 000000               MOV     #NOP,R2
1113 024374                     10$:
1114 024374 004737 053530               JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
     024400 000404                      BR      20$             ;GO TO 20$ IF NO ERROR
     024402 000240                      NOP                     ;RETURN HERE IF ERROR
     024404 104000                      EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
     024406 000137 025106               JMP     110$            ;GO TO 110$ IF ERROR
1115 024412                     20$:
1116
1117                            ;SET DIAGNOSTIC MODE AND DIAGNOSTIC UNIT READY, DIAGNOSTIC ON
1118                            ;CYLINDER
1119 024412 112737 000024 001543        MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     024420 112737 000200 001544        MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     024426 012737 000001 001434        MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
     024434 004737 044424               JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     024440 000404                      BR      30$             ;GO TO 30$ IF NO ERROR
     024442 000240                      NOP                     ;RETURN HERE IF ERROR
     024444 104000                      EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     024446 000137 025106               JMP     110$            ;GO TO 110$ IF ERROR
1120 024452                     30$:
```

```
1121  024452  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
      024460  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      024466  012737  001401  001434          MOV     #DMD.MUR!MOC,RMMR1O     ;SET RMMR1 OUTPUT BUFFER = DMD'MUR.MOC
      024474  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
      024500  000404                          BR      35$             ;GO TO 35$ IF NO ERROR
      024502  000240                          NOP                     ;RETURN HERE IF ERROR
      024504  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      024506  000137  025106                  JMP     110$            ;GO TO 110$ IF ERROR
1122  024512                          35$:
1123
1124                                  ;SET VOLUME VALID BEFORE ISSUING FUNCTION CODE BEING TESTED
1125  024512  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
      024520  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
      024526  012737  000023  001410          MOV     #PAKACK!GO,RMCS1O       ;SET RMCS1 OUTPUT BUFFER = PAKACK!GO
      024534  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
      024540  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
      024542  000240                          NOP                     ;RETURN HERE IF ERROR
      024544  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
      024546  000137  025106                  JMP     110$            ;GO TO 110$ IF ERROR
1126  024552                          40$:
1127  024552  012737  000001  001410          MOV     #GO,RMCS1O
1128  024560  050237  001410                  BIS     R2,RMCS1O       ;WRITE FUNCTION CODE IN BUFFER
1129  024564  012737  103640  001414          MOV     #BUFONE,RMBAO   ;DUMMY BUS ADDRESS
1130  024572  012737  177777  001412          MOV     #-1,RMWCO       ;DUMMY WORD COUNT
1131  024600  012737  177777  001416          MOV     #-1,RMDAO       ;USE INVALID DISK ADDRESS
1132  024606  012737  177777  001444          MOV     #-1,RMDCO       ;USE INVALID CYLINDER ADDRESS
1133  024614  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
1134  024622  012703  001543                  MOV     #PUTINX,R3      ;WRITE REGISTER INDEX TABLE
1135  024626  112723  000004                  MOVB    #RMBA,(R3)+
1136  024632  112723  000002                  MOVB    #RMWC,(R3)+
1137  024636  112723  000032                  MOVB    #RMOF,(R3)+
1138  024642  112723  000006                  MOVB    #RMDA,(R3)+
1139  024646  112723  000034                  MOVB    #RMDC,(R3)+
1140  024652  112723  000000                  MOVB    #RMCS1,(R3)+
1141  024656  112713  000200                  MOVB    #200,(R3)
1142  024662  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      024666  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
      024670  000240                          NOP                     ;RETURN HERE IF ERROR
      024672  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      024674  000137  025106                  JMP     110$            ;GO TO 110$ IF ERROR
1143  024700  004737  044766          50$:    JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
1144
1145  024704  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      024710  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
      024712  000240                          NOP                     ;RETURN HERE IF ERROR
      024714  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      024716  000137  025106                  JMP     110$            ;GO TO 110$ IF ERROR
1146  024722                          60$:
1147  024722  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      024726  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
      024730  000240                          NOP                     ;RETURN HERE IF ERROR
      024732  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      024734  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      024736  000137  025106                  JMP     110$            ;GO TO 110$ IF ERROR
1148  024742  016237  067046  001140  70$:    MOV     FNCDTB(R2),$GDDAT       ;GET ENTRY FROM FUNCTION CODE TABLE
1149  024750  042737  175777  001140          BIC     #^C1AE,$GDDAT
1150  024756  013737  001350  001142          MOV     RMER1I,$BDDAT   ;BAD DATA FOR TYPEOUT
```

```
1151 024764  042737  175777  001142          BIC     #^CIAE,$BDDAT
1152 024772  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS IAE STATUS CORRECT?
1153 025000  001402                          BEQ     80$             ;YES!!
1154 025002  104217                          EMT     217
1155 025004  000440                          BR      110$
1156 025006  016237  067046  001140  80$:    MOV     FNCDTB(R2),$GDDAT
1157 025014  032737  040000  001346          BIT     #ERR,RMDSI      ;WAS AN ERROR DETECTED??
1158 025022  001403                          BEQ     90$             ;NO!!
1159 025024  052737  100000  001140          BIS     #ATA,$GDDAT     ;YES - ATA SHOULD BE ON
1160 025032  042737  077777  001140  90$:    BIC     #^CATA,$GDDAT
1161 025040  013737  001346  001142          MOV     RMDSI,$BDDAT    ;GET DRIVE'S ATTENTION
1162 025046  042737  077777  001142          BIC     #^CATA,$BDDAT
1163 025054  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA STATUS OK??
1164 025062  001402                          BEQ     100$            ;YES!!
1165 025064  104255                          EMT     255
1166 025066  000407                          BR      110$
1167 025070  062702  000002          100$:   ADD     #2,R2           ;GO TO NEXT FUNCTION CODE
1168 025074  022702  000076                  CMP     #ILF76,R2       ;DONE??
1169 025100  103402                          BLO     110$            ;YES !!
1170 025102  000137  024374                  JMP     10$             ;TEST NEXT FUNCTION
1171 025106                          110$:
1172
1173                                 ;;*****************************************************************
                                     ;*TEST 36       WRITE LOCK ERROR TEST
                                     ;;*****************************************************************
     025106                          TST36:
     025106  000004                          SCOPE                   ;SCOPE CALL
     025110  000240                          NOP                     ;START OF TEST
     025112  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     025116  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     025122  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     025126  012737  000036  001226          MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1174
1175 025134  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     025140  040000                          .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                             ;CLEAR CONTROLLER & SELECT DEVICE
     025142  000404                          BR      5$              ;GO TO 5$ IF NO ERROR
     025144  000240                          NOP                     ;RETURN HERE IF ERROR
     025146  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     025150  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
1176 025154                          5$:
1177 025154  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS
1178 025160  012702  000000                  MOV     #NOP,R2
1179 025164                          10$:
1180 025164  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
     025170  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
     025172  000240                          NOP                     ;RETURN HERE IF ERROR
     025174  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
     025176  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
1181 025202                          20$:
1182
1183                                 ;SET DIAGNOSTIC MODE AND DIAGNOSTIC UNIT READY, DIAGNOSTIC ON
1184                                 ;CYLINDER, AND DIAGNOSTIC WRITE LOCK
1185 025202  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     025210  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     025216  012737  000001  001434          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER - DMD
     025224  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
```

```
        025230  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        025232  000240                          NOP                     ;RETURN HERE IF ERROR
        025234  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        025236  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
  1186  025242                          30$:
  1187  025242  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
        025250  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        025256  012737  001411  001434          MOV     #DMD!MUR!MOC!MWP,RMMR10 ;SET RMMR1 OUTPUT BUFFER = DMD.MUR!MOC!MWP
        025264  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
        025270  000404                          BR      35$             ;GO TO 35$ IF NO ERROR
        025272  000240                          NOP                     ;RETURN HERE IF ERROR
        025274  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        025276  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
  1188  025302                          35$:
  1189
  1190                                  ;SET VOLUME VALID BEFORE ISSUING FUNCTION CODE BEING TESTED
  1191  025302  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
        025310  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        025316  012737  000023  001410          MOV     #PAKACK!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = PAKACK!GO
        025324  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
        025330  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        025332  000240                          NOP                     ;RETURN HERE IF ERROR
        025334  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        025336  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
  1192  025342                          40$:
  1193  025342  012737  000001  001410          MOV     #GO,RMCS10
  1194  025350  050237  001410                  BIS     R2,RMCS10       ;WRITE FUNCTION CODE IN BUFFER
  1195  025354  012737  103640  001414          MOV     #BUFONE,RMBA0   ;DUMMY BUS ADDRESS
  1196  025362  012737  177777  001412          MOV     #-1,RMWC0       ;DUMMY WORD COUNT
  1197  025370  005037  001416                  CLR     RMDA0           ;CLEAR DISK ADDRESS
  1198  025374  005037  001444                  CLR     RMDC0           ;CLEAR CYLINDER ADDRESS
  1199  025400  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
  1200  025406  012703  001543                  MOV     #PUTINX,R3      ;WRITE REGISTER INDEX TABLE
  1201  025412  112723  000004                  MOVB    #RMBA,(R3)+
  1202  025416  112723  000002                  MOVB    #RMWC,(R3)+
  1203  025422  112723  000032                  MOVB    #RMOF,(R3)+
  1204  025426  112723  000006                  MOVB    #RMDA,(R3)+
  1205  025432  112723  000034                  MOVB    #RMDC,(R3)+
  1206  025436  112723  000000                  MOVB    #RMCS1,(R3)+
  1207  025442  112713  000200                  MOVB    #200,(R3)
  1208  025446  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        025452  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        025454  000240                          NOP                     ;RETURN HERE IF ERROR
        025456  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        025460  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
  1209  025464  004737  044766          50$:    JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
  1210
  1211  025470  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        025474  000404                          BR      60$             ;GO TO 60$ IF NO ERROR
        025476  000240                          NOP                     ;RETURN HERE IF ERROR
        025500  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        025502  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
  1212  025506                          60$:
  1213  025506  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        025512  000405                          BR      70$             ;GO TO 70$ IF NO ERROR
        025514  000240                          NOP                     ;RETURN HERE IF ERROR
        025516  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
```

```
025520  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
025522  000137  025672                  JMP     110$            ;GO TO 110$ IF ERROR
1214 025526  016237  067046  001140  70$:    MOV     FNCDTB(R2),$GDDAT       ;GET ENTRY FROM FUNCTION CODE TABLE
1215 025534  042737  173777  001140          BIC     #^CWLE,$GDDAT
1216 025542  013737  001350  001142          MOV     RMER11,$BDDAT   ;BAD DATA FOR TYPEOUT
1217 025550  042737  173777  001142          BIC     #^CWLE,$BDDAT
1218 025556  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS WLE STATUS CORRECT?
1219 025564  001402                          BEQ     80$             ;YES!.
1220 025566  104220                          EMT     220
1221 025570  000440                          BR      110$
1222 025572  016237  067046  001140  80$:    MOV     FNCDTB(R2),$GDDAT
1223 025600  032737  040000  001346          BIT     #ERR,RMDSI      ;WAS AN ERROR DETECTED??
1224 025606  001403                          BEQ     90$             ;NO!!
1225 025610  052737  100000  001140          BIS     #ATA,$GDDAT     ;YES - ATA SHOULD BE ON
1226 025616  042737  077777  001140  90$:    BIC     #^CATA,$GDDAT
1227 025624  013737  001346  001142          MOV     RMDSI,$BDDAT    ;GET DRIVE'S ATTENTION
1228 025632  042737  077777  001142          BIC     #^CATA,$BDDAT
1229 025640  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA STATUS OK??
1230 025646  001402                          BEQ     100$            ;YES!!
1231 025650  104255                          EMT     255
1232 025652  000407                          BR      110$
1233 025654  062702  000002          100$:   ADD     #2,R2           ;GO TO NEXT FUNCTION CODE
1234 025660  022702  000076                  CMP     #ILF76,R2       ;DONE??
1235 025664  103402                          BLO     110$            ;YES !!
1236 025666  000137  025164                  JMP     10$             ;TEST NEXT FUNCTION
1237 025672                          110$:
1238
1239                                 ;;*****************************************************************
                                     ;*TEST 37       ERROR ABORT TESTS
                                     ;;*****************************************************************
     025672                          TST37:
     025672  000004                          SCOPE                   ;SCOPE CALL
     025674  000240                          NOP                     ;START OF TEST
     025676  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     025702  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     025706  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     025712  012737  000037  001226          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1240
1241 025720  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     025724  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
                                                                     ;VERIFY CONTROLLER CLEAR OPERATION
                                                                     ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                     ;VERIFY PACK ACKNOWLEDGE
                                                                     ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                     ;VERIFY RECALIBRATION
     025726  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
     025730  000240                          NOP                     ;RETURN HERE IF ERROR
     025732  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     025734  000137  026224                  JMP     80$             ;GO TO 80$ IF ERROR
1242 025740                          10$:
1243
1244                                 ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     025740  004737  044070                  JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1245 025744  012702  000000                  MOV     #NOP,R2         ;R2 - FUNCTION CODE
1246 025750                          20$:
1247 025750  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
```

```
        025754  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        025756  000240                          NOP                     ;RETURN HERE IF ERROR
        025760  104000                          EMT                     ;ERROR # DEFINED BY CNTCLR SUBROUTINE
        025762  000137  026224                  JMP     80$             ;GO TO 80$ IF ERROR
1248    025766                          30$:
1249    025766  112737  000014  001543          MOVB    #RMER1,PUTINX   ;SETUP PUT INDEX TABLE
        025774  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        026002  012737  040000  001424          MOV     #UNS,RMER10     ;SET RMER1 OUTPUT BUFFER = UNS
        026010  004737  044424                  JSR     PC,PUT          ;GO WRITE RMER1 VIA PUT SUBROUTINE
        026014  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        026016  000240                          NOP                     ;RETURN HERE IF ERROR
        026020  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        026022  000137  026224                  JMP     80$             ;GO TO 80$ IF ERROR
1250    026026                          40$:
1251    026026  012737  000001  001410          MOV     #GO,RMCS10
1252    026034  050237  001410                  BIS     R2,RMCS10       ;WRITE FUNCTION CODE IN BUFFER
1253    026040  012737  103640  001414          MOV     #BUFONE,RMBA0   ;DUMMY BUS ADDRESS
1254    026046  012737  177777  001412          MOV     #-1,RMWC0       ;DUMMY WORD COUNT
1255    026054  012737  000000  001416          MOV     #0,RMDA0        ;CLEAR DISK ADDRESS
1256    026062  012737  000000  001444          MOV     #0,RMDC0        ;CLEAR CYLINDER ADDRESS
1257    026070  012737  010000  001442          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
1258    026076  012703  001543                  MOV     #PUTINX,R3      ;WRITE REGISTER INDEX TABLE
1259    026102  112723  000004                  MOVB    #RMBA,(R3)+
1260    026106  112723  000002                  MOVB    #RMWC,(R3)+
1261    026112  112723  000032                  MOVB    #RMOF,(R3)+
1262    026116  112723  000006                  MOVB    #RMDA,(R3)+
1263    026122  112723  000034                  MOVB    #RMDC,(R3)+
1264    026126  112723  000000                  MOVB    #RMCS1,(R3)+
1265    026132  112713  000200                  MOVB    #200,(R3)
1266    026136  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        026142  000402                          BR      45$             ;GO TO 45$ IF NO ERROR
        026144  000240                          NOP                     ;RETURN HERE IF ERROR
        026146  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
1267    026150                          45$:
1268
1269                                    ;WAIT FOR COMMAND TO COMPLETE
        026150  004737  044766                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1270
1271    026154  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        026160  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        026162  000240                          NOP                     ;RETURN HERE IF ERROR
        026164  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        026166  000137  026224                  JMP     80$             ;GO TO 80$ IF ERROR
1272    026172                          50$:
1273    026172  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        026176  000405                          BR      60$             ;GO TO 60$ IF NO ERROR
        026200  000240                          NOP                     ;RETURN HERE IF ERROR
        026202  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        026204  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        026206  000137  026224                  JMP     80$             ;GO TO 80$ IF ERROR
1274    026212                          60$:
1275    026212                          70$:
1276    026212  062702  000002                  ADD     #2,R2           ;ADVANCE FUNCTION CODE
1277    026216  022702  000076                  CMP     #ILF76,R2       ;DONE ALL COMMANDS
1278    026222  103252                          BHIS    20$             ;NO !!
1279    026224                          80$:
1280
```

```
1281                                    ;;*****************************************************************
                                        ;*TEST 40       RMR TEST
                                        ;;*****************************************************************
     026224                             TST40:
     026224  000004                             SCOPE                   ;SCOPE CALL
     026226  000240                             NOP                     ;START OF TEST
     026230  012706  001100                     MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     026234  013700  001276                     MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     026240  013701  001464                     MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     026244  012737  000040  001226             MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1282
1283 026252  004737  043150                     JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     026256  040000                             .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
     026260  000404                             BR      10$             ;GO TO 10$ IF NO ERROR
     026262  000240                             NOP                     ;RETURN HERE IF ERROR
     026264  104000                             EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     026266  000137  026664                     JMP     100$            ;GO TO 100$ IF ERROR
1284 026272                             10$:
1285
1286                                    ;SETUP GET INDEX TABLE TO READ ALL REGISTERS
     026272  004737  044070                     JSR     PC,GETSTS       ;GO TO GETSTS SUBROUTINE
1287 026276                             20$:
1288 026276  112737  000024  001543             MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     026304  112737  000200  001544             MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026312  012737  000001  001434             MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
     026320  004737  044424                     JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     026324  000404                             BR      30$             ;GO TO 30$ IF NO ERROR
     026326  000240                             NOP                     ;RETURN HERE IF ERROR
     026330  104000                             EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026332  000137  026664                     JMP     100$            ;GO TO 100$ IF ERROR
1289 026336                             30$:
1290 026336  112737  000024  001543             MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     026344  112737  000200  001544             MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026352  012737  001401  001434             MOV     #DMD!MUR!MOC,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD!MUR!MOC
     026360  004737  044424                     JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     026364  000404                             BR      40$             ;GO TO 40$ IF NO ERROR
     026366  000240                             NOP                     ;RETURN HERE IF ERROR
     026370  104000                             EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026372  000137  026664                     JMP     100$            ;GO TO 100$ IF ERROR
1291 026376                             40$:
1292 026376  112737  000000  001543             MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
     026404  112737  000200  001544             MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026412  012737  000023  001410             MOV     #PAKACK!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = PAKACK!GO
     026420  004737  044424                     JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     026424  000404                             BR      50$             ;GO TO 50$ IF NO ERROR
     026426  000240                             NOP                     ;RETURN HERE IF ERROR
     026430  104000                             EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026432  000137  026664                     JMP     100$            ;GO TO 100$ IF ERROR
1293 026436                             50$:
1294 026436  004737  044154                     JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     026442  000404                             BR      60$             ;GO TO 60$ IF NO ERROR
     026444  000240                             NOP                     ;RETURN HERE IF ERROR
     026446  104000                             EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     026450  000137  026664                     JMP     100$            ;GO TO 100$ IF ERROR
1295 026454                             60$:
1296 026454  000240                             NOP
```

```
1297 026456  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     026464  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026472  012737  041401  001434          MOV     #DMD!MUR!MOC!DBEN,RMMR10        ;SET RMMR1 OUTPUT BUFFER = DMD!MUR!MOC!DBEN
     026500  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
     026504  000404                          BR      70$             ;GO TO 70$ IF NO ERROR
     026506  000240                          NOP                     ;RETURN HERE IF ERROR
     026510  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026512  000137  026664                  JMP     100$            ;GO TO 100$ IF ERROR
1298 026516                          70$:
1299 026516  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
     026524  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026532  012737  000001  001410          MOV     #NOP!GO,RMCS10  ;SET RMCS1 OUTPUT BUFFER = NOP!GO
     026540  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     026544  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
     026546  000240                          NOP                     ;RETURN HERE IF ERROR
     026550  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026552  000137  026664                  JMP     100$            ;GO TO 100$ IF ERROR
1300 026556                          80$:
1301 026556  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;SETUP PUT INDEX TABLE
     026564  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     026572  012737  000015  001410          MOV     #OFFSET!GO,RMCS10       ;SET RMCS1 OUTPUT BUFFER = OFFSET!GO
     026600  004737  044424                  JSR     PC,PUT          ;GO WRITE RMCS1 VIA PUT SUBROUTINE
     026604  000404                          BR      85$             ;GO TO 85$ IF NO ERROR
     026606  000240                          NOP                     ;RETURN HERE IF ERROR
     026610  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
     026612  000137  026664                  JMP     100$            ;GO TO 100$ IF ERROR
1302 026616                          85$:
1303 026616  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     026622  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
     026624  000240                          NOP                     ;RETURN HERE IF ERROR
     026626  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     026630  000137  026664                  JMP     100$            ;GO TO 100$ IF ERROR
1304 026634                          90$:
1305 026634  000240                          NOP
1306 026636  032737  000004  001350          BIT     #RMR,RMER11     ;IS RMR SET ??
1307 026644  001007                          BNE     100$            ;YES !!
1308 026646  012737  000004  001140          MOV     #RMR,$GDDAT     ;EXPECTED STATUS
1309 026654  013737  001350  001142          MOV     RMER11,$BDDAT   ;RECEIVED STATUS
1310 026662  104222                          EMT     222
1311 026664                          100$:
1312
1313                                 ;;*****************************************************************
                                     ;*TEST 41        PARITY ERROR TEST
                                     ;;*****************************************************************
     026664                          TST41:
     026664  000004                          SCOPE                   ;SCOPE CALL
     026666  000240                          NOP                     ;START OF TEST
     026670  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     026674  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     026700  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     026704  012737  000041  001226          MOV     #41,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1314
1315 026712  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     026716  040000                          .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                     ;CLEAR CONTROLLER & SELECT DEVICE
     026720  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
     026722  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
                026724  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                026726  000137  027104                  JMP     70$             ;GO TO 70$ IF ERROR
     1316       026732                          10$:
     1317       026732  012702  000001                  MOV     #1,R2           ;R2 = DATA PATTERN
     1318       026736                          20$:
     1319       026736  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
                026742  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
                026744  000240                          NOP                     ;RETURN HERE IF ERROR
                026746  104000                          EMT                     ;ERROR # DEFINED BY CNTCLR SUBROUTINE
                026750  000137  027104                  JMP     70$             ;GO TO 70$ IF ERROR
     1320       026754                          30$:
     1321       026754  111103                          MOVB    (R1),R3         ;SETUP RMCS2
     1322       026756  042703  177770                  BIC     #^CUNTMSK,R3
     1323       026762  052703  000020                  BIS     #PAT,R3
     1324       026766  010337  001420                  MOV     R3,RMCS20       ;OUTPUT VALUE TO RMCS2
     1325       026772  010237  001416                  MOV     R2,RMDA0        ;VALUE TO RMDA
     1326       026776  012703  001543                  MOV     #PUTINX,R3      ;WRITE REGISTER OUTPUT INDEX
     1327       027002  112723  000010                  MOVB    #RMCS2,(R3)+
     1328       027006  112723  000006                  MOVB    #RMDA,(R3)+
     1329       027012  112723  000200                  MOVB    #200,(R3)+
     1330
     1331       027016  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                027022  000402                          BR      40$             ;GO TO 40$ IF NO ERROR
                027024  000240                          NOP                     ;RETURN HERE IF ERROR
                027026  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     1332       027030                          40$:
     1333       027030  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                027034  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
                027036  000240                          NOP                     ;RETURN HERE IF ERROR
                027040  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                027042  000137  027104                  JMP     70$             ;GO TO 70$ IF ERROR
     1334       027046                          50$:
     1335       027046  032737  000010  001350          BIT     #PAR,RMER1I     ;IS PARITY ERROR SET ??
     1336       027054  001011                          BNE     60$             ;YES !!
     1337       027056  012737  000010  001140          MOV     #PAR,$GDDAT     ;EXPECTED STATUS
     1338       027064  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
     1339       027072  010237  001174                  MOV     R2,$TMP0        ;DATA PATTERN
     1340       027076  104223                          EMT     223
     1341       027100                          60$:
     1342       027100  006302                          ASL     R2              ;ADVANCE DATA PATTERN
     1343       027102  001315                          BNE     20$             ;BRANCH IF NOT DONE
     1344       027104                          70$:
     1345
     1346                                       ;;*******************************************************************
                                                ;*TEST 42        ILLEGAL REGISTER TEST
                                                ;;*******************************************************************
                027104                          TST42:
                027104  000004                          SCOPE                   ;SCOPE CALL
                027106  000240                          NOP                     ;START OF TEST
                027110  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                027114  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                027120  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                027124  012737  000042  001226          MOV     #42,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     1347
     1348       027132  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                027136  040000                          .WORD   040000          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
```

```
        027140  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
        027142  000240                          NOP                     ;RETURN HERE IF ERROR
        027144  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        027146  000137  027432                  JMP     120$            ;GO TO 120$ IF ERROR
  1349  027152                          10$:
  1350  027152  005005                          CLR     R5              ;R5 = EXPECTED STATUS
  1351  027154  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS
  1352  027160  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
  1353  027164  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
  1354  027170  012737  027430  000004          MOV     #110$,ERRVEC    ;SETUP FOR BUS TIMEOUT
  1355  027176  012737  000300  000006          MOV     #PR6,ERRVEC+2
  1356  027204  005002                          CLR     R2              ;R2 = REGISTER INDEX
  1357  027206                          20$:
  1358  027206  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
        027212  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        027214  000240                          NOP                     ;RETURN HERE IF ERROR
        027216  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
        027220  000137  027344                  JMP     70$             ;GO TO 70$ IF ERROR
  1359  027224  010003                  30$:    MOV     R0,R3           ;R3 = REGISTER ADDRESS
  1360  027226  060203                          ADD     R2,R3
  1361  027230  005013                          CLR     (R3)            ;CLEAR THE REGISTER
  1362  027232  004737  052056                  JSR     PC,DEVSEL       ;GO SELECT DEVICE
        027236  000404                          BR      35$             ;GO TO 35$ IF NO ERROR
        027240  000240                          NOP                     ;RETURN HERE IF ERROR
        027242  104000                          EMT                     ;ERROR # DEFINED BY DEVSEL SUBROUTINE
        027244  000137  027432                  JMP     120$            ;GO TO 120$ IF ERROR
  1363  027250                          35$:
  1364  027250  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        027254  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        027256  000240                          NOP                     ;RETURN HERE IF ERROR
        027260  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        027262  000137  027344                  JMP     70$             ;GO TO 70$ IF ERROR
  1365  027266                          40$:
  1366  027266  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        027272  000405                          BR      50$             ;GO TO 50$ IF NO ERROR
        027274  000240                          NOP                     ;RETURN HERE IF ERROR
        027276  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        027300  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        027302  000137  027344                  JMP     70$             ;GO TO 70$ IF ERROR
  1367  027306  010537  001140          50$:    MOV     R5,$GDDAT
  1368  027312  013737  001350  001142          MOV     RMER11,$BDDAT   ;GET DRIVE'S ILR STATUS
  1369  027320  042737  177775  001142          BIC     #^CILR,$BDDAT
  1370  027326  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ILR STATUS OK??
  1371  027334  001403                          BEQ     70$             ;YES!!
  1372  027336  010237  001174                  MOV     R2,$TMP0        ;SAVE R2 FOR ERROR DATA
  1373  027342  104256                          EMT     256
  1374                                  ;70$:   ADD     #2,R2
  1375                                  ;       CMP     #RMBAE,R2       ;IS NEXT REGISTER RMBAE??
  1376                                  ;       BNE     80$             ;NO!!
  1377                                  ;       MOV     #A16!A17,RMCS1(R0)      ;SET A16 & A17
  1378                                  ;       MOV     RMBAE(R0),R3    ;IS THIS AN RH70??
  1379                                  ;       BIC     #^C<BIT0!BIT1>,R3       ;CLEAR ALL BUT ADDRESS BITS
  1380                                  ;       CMP     #BIT0!BIT1,R3   ;ARE ADDRESS BITS ON ??
  1381                                  ;       BEQ     100$            ;YES
  1382                                  ;       MOV     #ILR,R5         ;NO - EXPECT ILR - 1
  1383                                  ;       BR      100$
  1384                                  ;80$:   CMP     #RMCS3+2,R2     ;SHOULD ILR BE SET??
```

```
                                        ;       BNE     90$             ;NO.!
                                        ;       MOV     #ILR,R5         ;YES.!
                                        ;       BR      100$
                                        ;90$:   CMP     #76,R2          ;DONE??
                                        ;       BLO     120$            ;YES!!
                                        ;100$:  BR      20$
                                        ;THE FOLLOWING CODING FOR HANDLE RH70
                                        ;REGISTER NUMBERS IS EITHER 22 OR 32
027344  062702  000002                  70$:    ADD     #2,R2           ;INCREMENT THE REGISTER ADDRESS
027350  022702  000050                          CMP     #50,R2          ;TIME TO CHECK THE EXTEND ADDRESS REG ?
027354  101314                                  BHI     20$             ;BRANCH IF NOT
027356  103420                                  BLO     80$             ;BRANCH IF ALREADY SET UP
027360  012705  000002                          MOV     #ILR,R5         ;EXCEPT ILR HAPPEND
027364  012760  001400  000000                  MOV     #A16!A17,RMCS1(R0)      ;SET EXTENDED ADDRESS BITS
027372  012702  000054                          MOV     #54,R2          ;SET ADDRESS TO 54,IF 22 REGISTERS
027376  016003  000050                          MOV     50(R0),R3       ;IS THIS 22 REG RH70 ?
027402  042703  177774                          BIC     #177774,R3      ;LEFT ONLY BIT 0 AND BIT 1
027406  022703  000003                          CMP     #BIT0!BIT1,R3   ;BIT 0 AND BIT 1 SET AT THE SAME TIME ?
027412  001402                                  BEQ     80$             ;BRANCH IF SO
027414  012702  000050                          MOV     #50,R2          ;OTHERWISE,SET ADDRESS TO 50
027420  022702  000074                  80$:    CMP     #74,R2          ;ALL REGISTERS CHECKED ?
027424  101402                                  BLOS    120$            ;BRANCH IF SO
027426  000667                                  BR      20$             ;NEXT REGISTER

027430  022626                          110$:   CMP     (SP)+,(SP)+
027432                                  120$:
027432  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
027436  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC

                                        ;;**********************************************************************
                                        ;*TEST 43       SEEK LAST CYLINDER
                                        ;;**********************************************************************
027442                                  TST43:
027442  000004                                  SCOPE                   ;SCOPE CALL
027444  000240                                  NOP                     ;START OF TEST
027446  012706  001100                          MOV     #STACK,SP       ;INITIALIZE STACK POINTER
027452  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
027456  013701  001464                          MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
027462  012737  000001  001206                  MOV     #1,$TIMES       ;;DO 1 ITERATION
027470  012737  000043  001226                  MOV     #43,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

027476                                  10$:
027476  004737  043150                          JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
027502  054130                                  .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
027504  000404                                  BR      80$             ;GO TO 80$ IF NO ERROR
027506  000240                                  NOP                     ;RETURN HERE IF ERROR
027510  104000                                  EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
027512  000137  027730                          JMP     140$            ;GO TO 140$ IF ERROR
027516                                  80$:
027516  012737  001466  001444                  MOV     #822.,RMDCO     ;LAST CYLINDER
027524  012737  000000  001416                  MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
```

```
1420 027532  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;LOAD SEEK COMMAND IN BUFFER.
1421 027540  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
1422 027544  112722  000034                  MOVB    #RMDC,(R2)+     ;WRITE REGISTER INDEX TABLE
1423 027550  112722  000006                  MOVB    #RMDA,(R2)+
1424 027554  112722  000000                  MOVB    #RMCS1,(R2)+
1425 027560  112722  000200                  MOVB    #200,(R2)+      ;WRITE TERMINATOR
1426 027564  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     027570  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
     027572  000240                          NOP                     ;RETURN HERE IF ERROR
     027574  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     027576  000137  027730                  JMP     140$            ;GO TO 140$ IF ERROR
1427 027602  004737  044070          90$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
1428 027606  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
1429
1430 027612  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     027616  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
     027620  000240                          NOP                     ;RETURN HERE IF ERROR
     027622  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     027624  000137  027730                  JMP     140$            ;GO TO 140$ IF ERROR
1431 027630                          100$:
1432 027630  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     027634  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
     027636  000240                          NOP                     ;RETURN HERE IF ERROR
     027640  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     027642  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027644  000137  027730                  JMP     140$            ;GO TO 140$ IF ERROR
1433 027650                          110$:
1434 027650  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     027654  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
     027656  000240                          NOP                     ;RETURN HERE IF ERROR
     027660  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     027662  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027664  000137  027730                  JMP     140$            ;GO TO 140$ IF ERROR
1435 027670                          120$:
1436 027670  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     027674  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
     027676  000010                          .WORD   DPE             ;MASK FOR RMER2
     027700  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
     027702  000240                          NOP                     ;RETURN HERE IF ERROR
     027704  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     027706  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     027710  000137  027730                  JMP     140$            ;GO TO 140$ IF ERROR
1437 027714                          130$:
1438 027714  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     027720  000403                          BR      140$            ;GO TO 140$ IF NO ERROR
     027722  000240                          NOP                     ;RETURN HERE IF ERROR
     027724  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     027726  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1439 027730                          140$:
1440
1441                                 ;;*****************************************************************
                                     ;*TEST 44        SEEK FIRST CYLINDER
                                     ;;*****************************************************************
     027730                          TST44:
     027730  000004                          SCOPE                   ;SCOPE CALL
     027732  000240                          NOP                     ;START OF TEST
     027734  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
```

```
                027740  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                027744  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                027750  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
                027756  012737  000044  001226          MOV     #44,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     1442
     1443 027764                                10$:
     1444 027764  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
          027770  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                          ;CLEAR CONTROLLER & SELECT DEVICE
                                                                          ;VERIFY CONTROLLER CLEAR OPERATION
                                                                          ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                          ;VERIFY PACK ACKNOWLEDGE
                                                                          ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                          ;VERIFY RECALIBRATION
          027772  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
          027774  000240                          NOP                     ;RETURN HERE IF ERROR
          027776  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
          030000  000137  030216                  JMP     140$            ;GO TO 140$ IF ERROR
     1445 030004                                80$:
     1446 030004  012737  000000  001444          MOV     #0.,RMDC0       ;CYLINDER = 0
     1447 030012  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
     1448 030020  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;LOAD SEEK COMMAND IN BUFFER
     1449 030026  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
     1450 030032  112722  000034                  MOVB    #RMDC,(R2)+     ;WRITE REGISTER INDEX TABLE
     1451 030036  112722  000006                  MOVB    #RMDA,(R2)+
     1452 030042  112722  000000                  MOVB    #RMCS1,(R2)+
     1453 030046  112722  000200                  MOVB    #200,(R2)+      ;WRITE TERMINATOR
     1454 030052  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
          030056  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
          030060  000240                          NOP                     ;RETURN HERE IF ERROR
          030062  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
          030064  000137  030216                  JMP     140$            ;GO TO 140$ IF ERROR
     1455 030070  004737  044070          90$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
     1456 030074  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
     1457
     1458 030100  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
          030104  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
          030106  000240                          NOP                     ;RETURN HERE IF ERROR
          030110  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
          030112  000137  030216                  JMP     140$            ;GO TO 140$ IF ERROR
     1459 030116                                100$:
     1460 030116  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
          030122  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
          030124  000240                          NOP                     ;RETURN HERE IF ERROR
          030126  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
          030130  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
          030132  000137  030216                  JMP     140$            ;GO TO 140$ IF ERROR
     1461 030136                                110$:
     1462 030136  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
          030142  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
          030144  000240                          NOP                     ;RETURN HERE IF ERROR
          030146  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
          030150  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
          030152  000137  030216                  JMP     140$            ;GO TO 140$ IF ERROR
     1463 030156                                120$:
     1464 030156  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
          030162  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
```

```
        030164  000010                          .WORD   DPE             ;MASK FOR RMER2
        030166  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
        030170  000240                          NOP                     ;RETURN HERE IF ERROR
        030172  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        030174  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        030176  000137  030216                  JMP     140$            ;GO TO 140$ IF ERROR
1465    030202                          130$:
1466    030202  004737  046004                 JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        030206  000403                          BR      140$            ;GO TO 140$ IF NO ERROR
        030210  000240                          NOP                     ;RETURN HERE IF ERROR
        030212  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        030214  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1467    030216                          140$:
1468
1469                                    ;;*******************************************************************
                                        ;*TEST 45        SEEK PRIME CYLINDERS
                                        ;;*******************************************************************
        030216                          TST45:
        030216  000004                          SCOPE                   ;SCOPE CALL
        030220  000240                          NOP                     ;START OF TEST
        030222  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        030226  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        030232  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        030236  012737  000002  001206          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
        030244  012737  000045  001226          MOV     #45,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1470
1471    030252  012737  000001  001444          MOV     #1,RMDC0        ;FIRST CYLINDER = 1
1472    030260                          10$:
1473    030260  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        030264  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        030266  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        030270  000240                          NOP                     ;RETURN HERE IF ERROR
        030272  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        030274  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1474    030300                          80$:
1475    030300  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
1476    030306  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;LOAD SEEK COMMAND IN BUFFER
1477    030314  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
1478    030320  112722  000034                  MOVB    #RMDC,(R2)+     ;WRITE REGISTER INDEX TABLE
1479    030324  112722  000006                  MOVB    #RMDA,(R2)+
1480    030330  112722  000000                  MOVB    #RMCS1,(R2)+
1481    030334  112722  000200                  MOVB    #200,(R2)+      ;WRITE TERMINATOR
1482    030340  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        030344  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        030346  000240                          NOP                     ;RETURN HERE IF ERROR
        030350  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        030352  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1483    030356                          90$:
1484    030356  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
1485    030362  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
1486
```

```
1487 030366  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     030372  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
     030374  000240                          NOP                     ;RETURN HERE IF ERROR
     030376  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     030400  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1488 030404                          100$:
1489 030404  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     030410  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
     030412  000240                          NOP                     ;RETURN HERE IF ERROR
     030414  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     030416  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     030420  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1490 030424                          110$:
1491 030424  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     030430  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
     030432  000240                          NOP                     ;RETURN HERE IF ERROR
     030434  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     030436  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     030440  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1492 030444                          120$:
1493 030444  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     030450  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
     030452  000010                          .WORD   DPE             ;MASK FOR RMER2
     030454  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
     030456  000240                          NOP                     ;RETURN HERE IF ERROR
     030460  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     030462  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     030464  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1494 030470                          130$:
1495 030470  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     030474  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
     030476  000240                          NOP                     ;RETURN HERE IF ERROR
     030500  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     030502  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     030504  000137  030530                  JMP     150$            ;GO TO 150$ IF ERROR
1496 030510                          140$:
1497 030510  006337  001444                  ASL     RMDCO           ;SHIFT TO NEXT PRIME CYLINDER
1498 030514  022737  001000  001444          CMP     #512.,RMDCO     ;IS THE TEST DONE??
1499 030522  103402                          BLO     150$            ;YES!!
1500 030524  000137  030260                  JMP     10$             ;GO DO NEXT CYLINDER
1501 030530                          150$:
1502
1503                                 ;;*******************************************************************
                                     ;*TEST 46        SEEK ZERO DIFFERENCE
                                     ;;*******************************************************************
     030530                          TST46:
     030530  000004                          SCOPE                   ;SCOPE CALL
     030532  000240                          NOP                     ;START OF TEST
     030534  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     030540  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     030544  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     030550  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
     030556  012737  000046  001226          MOV     #46,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1504
1505 030564                          10$:
1506 030564  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     030570  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
```

```
                030572  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                030574  000240                          NOP                     ;RETURN HERE IF ERROR
                030576  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                030600  000137  031036                  JMP     170$            ;GO TO 170$ IF ERROR
1507            030604                          80$:
1508            030604  012703  000003                  MOV     #3,R3           ;R3 = NUMBER OF SEEKS
1509            030610  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
1510            030616  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
1511            030624  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;FUNCTION CODE FOR SEEK
1512            030632  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER INDEX
1513            030636  112722  000006                  MOVB    #RMDA,(R2)+     ;WRITE REGISTER INDEX TABLE
1514            030642  112722  000034                  MOVB    #RMDC,(R2)+
1515            030646  112722  000000                  MOVB    #RMCS1,(R2)+
1516            030652  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
1517            030656                          90$:
1518            030656  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                030662  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
                030664  000240                          NOP                     ;RETURN HERE IF ERROR
                030666  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                030670  000137  031036                  JMP     170$            ;GO TO 170$ IF ERROR
1519            030674  004737  044070          100$:   JSR     PC,GETSTS       ;SETUP FOR READING STATUS
1520            030700  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
1521
1522            030704  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                030710  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
                030712  000240                          NOP                     ;RETURN HERE IF ERROR
                030714  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                030716  000137  031036                  JMP     170$            ;GO TO 170$ IF ERROR
1523            030722                          120$:
1524            030722  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                030726  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
                030730  000240                          NOP                     ;RETURN HERE IF ERROR
                030732  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                030734  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                030736  000137  031036                  JMP     170$            ;GO TO 170$ IF ERROR
1525            030742                          130$:
1526            030742  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
                030746  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
                030750  000240                          NOP                     ;RETURN HERE IF ERROR
                030752  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
                030754  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                030756  000137  031036                  JMP     170$            ;GO TO 170$ IF ERROR
1527            030762                          140$:
1528            030762  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
                030766  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
                030770  000010                          .WORD   DPE             ;MASK FOR RMER2
                030772  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                030774  000240                          NOP                     ;RETURN HERE IF ERROR
                030776  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
                031000  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                031002  000137  031036                  JMP     170$            ;GO TO 170$ IF ERROR
```

```
1529 031006                                  150$:
1530 031006  004737  046004                        JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     031012  000405                                BR      160$            ;GO TO 160$ IF NO ERROR
     031014  000240                                NOP                     ;RETURN HERE IF ERROR
     031016  104000                                EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     031020  004736                                JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031022  000137  031036                        JMP     170$            ;GO TO 170$ IF ERROR
1531 031026                                  160$:
1532 031026  005303                                DEC     R3              ;DONE ALL SEEKS??
1533 031030  001402                                BEQ     170$            ;YES!!
1534 031032  000137  030656                        JMP     90$             ;NO - GO DO NEXT SEEK
1535 031036                                  170$:
1536
1537                                         ;;*******************************************************************
                                             ;*TEST 47       SEEK MAXIMUM DIFFERENCE FORWARD
                                             ;;*******************************************************************
     031036                                  TST47:
     031036  000004                                SCOPE                   ;SCOPE CALL
     031040  000240                                NOP                     ;START OF TEST
     031042  012706  001100                        MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     031046  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     031052  013701  001464                        MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     031056  012737  000002  001206                MOV     #2,$TIMES       ;;DO 2 ITERATIONS
     031064  012737  000047  001226                MOV     #47,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1538
1539 031072                                  10$:
1540 031072  004737  043150                        JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     031076  054130                                .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                           ;CLEAR CONTROLLER & SELECT DEVICE
                                                                           ;VERIFY CONTROLLER CLEAR OPERATION
                                                                           ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                           ;VERIFY PACK ACKNOWLEDGE
                                                                           ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                           ;VERIFY RECALIBRATION
     031100  000404                                BR      80$             ;GO TO 80$ IF NO ERROR
     031102  000240                                NOP                     ;RETURN HERE IF ERROR
     031104  104000                                EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     031106  000137  031350                        JMP     160$            ;GO TO 160$ IF ERROR
1541 031112                                  80$:
1542 031112  012737  000000  001444                MOV     #0,RMDC0        ;SEEK TO CYLINDER 0
1543 031120  012737  000000  001416                MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
1544 031126  012737  000005  001410                MOV     #SEEK!GO,RMCS10 ;FUNCTION CODE FOR SEEK
1545 031134  012702  001543                        MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
1546 031140  112722  000006                        MOVB    #RMDA,(R2)+     ;WRITE REGISTER INDEX TABLE
1547 031144  112722  000034                        MOVB    #RMDC,(R2)+
1548 031150  112722  000000                        MOVB    #RMCS1,(R2)+
1549 031154  112722  000200                        MOVB    #200,(R2)+      ;STEUP TERMINATOR BYTE IN TABLE
1550 031160                                  90$:
1551 031160  004737  044424                        JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031164  000404                                BR      100$            ;GO TO 100$ IF NO ERROR
     031166  000240                                NOP                     ;RETURN HERE IF ERROR
     031170  104000                                EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     031172  000137  031350                        JMP     160$            ;GO TO 160$ IF ERROR
1552 031176                                  100$:
1553 031176  004737  044070                        JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
1554 031202  004737  044766                        JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
1555
```

```
1556 031206  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     031212  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
     031214  000240                          NOP                     ;RETURN HERE IF ERROR
     031216  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     031220  000137  031350                  JMP     160$            ;GO TO 160$ IF ERROR
1557 031224                          110$:
1558 031224  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     031230  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
     031232  000240                          NOP                     ;RETURN HERE IF ERROR
     031234  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     031236  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031240  000137  031350                  JMP     160$            ;GO TO 160$ IF ERROR
1559 031244                          120$:
1560 031244  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     031250  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
     031252  000240                          NOP                     ;RETURN HERE IF ERROR
     031254  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     031256  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031260  000137  031350                  JMP     160$            ;GO TO 160$ IF ERROR
1561 031264                          130$:
1562 031264  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     031270  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
     031272  000010                          .WORD   DPE             ;MASK FOR RMER2
     031274  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
     031276  000240                          NOP                     ;RETURN HERE IF ERROR
     031300  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     031302  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031304  000137  031350                  JMP     160$            ;GO TO 160$ IF ERROR
1563 031310                          140$:
1564 031310  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     031314  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
     031316  000240                          NOP                     ;RETURN HERE IF ERROR
     031320  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     031322  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031324  000137  031350                  JMP     160$            ;GO TO 160$ IF ERROR
1565 031330                          150$:
1566 031330  005737  001444                  TST     RMDCO           ;IS TEST DONE ?
1567 031334  001005                          BNE     160$            ;YES!!
1568 031336  012737  001466  001444          MOV     #822.,RMDCO     ;NO - SEEK TO LAST CYLINDER
1569 031344  000137  031160                  JMP     90$
1570 031350                          160$:
1571
1572                                 ;;*****************************************************************
                                     ;*TEST 50       SEEK ADJACENT FORWARD
                                     ;;*****************************************************************
     031350                          TST50:
     031350  000004                          SCOPE                   ;SCOPE CALL
     031352  000240                          NOP                     ;START OF TEST
     031354  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     031360  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     031364  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     031370  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
     031376  012737  000050  001226          MOV     #50,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1573
1574 031404                          10$:
1575 031404  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     031410  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
```

```
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
     031412  000404                          BR      80$       ;GO TO 80$ IF NO ERROR
     031414  000240                          NOP               ;RETURN HERE IF ERROR
     031416  104000                          EMT               ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     031420  000137  031660                  JMP     160$      ;GO TO 160$ IF ERROR
1576 031424                          80$:
1577 031424  012737  000000  001444          MOV     #0,RMDCO  ;CYLINDER = 0
1578 031432  012737  000000  001416          MOV     #0,RMDAO  ;TRACK = 0, SECTOR = 0
1579 031440  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;FUNCTION CODE FOR SEEK
1580 031446  012702  001543                  MOV     #PUTINX,R2 ;R2 POINTS TO REGISTER INDEX
1581 031452  112722  000034                  MOVB    #RMDC,(R2)+ ;WRITE REGISTER INDEX TABLE
1582 031456  112722  000006                  MOVB    #RMDA,(R2)+
1583 031462  112722  000000                  MOVB    #RMCS1,(R2)+
1584 031466  112722  000200                  MOVB    #200,(R2)+ ;TERMINATE REGISTER TABLE
1585 031472                          90$:
1586 031472  004737  044424                  JSR     PC,PUT    ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     031476  000404                          BR      100$      ;GO TO 100$ IF NO ERROR
     031500  000240                          NOP               ;RETURN HERE IF ERROR
     031502  104000                          EMT               ;ERROR # DEFINED BY PUT SUBROUTINE
     031504  000137  031660                  JMP     160$      ;GO TO 160$ IF ERROR
1587 031510  004737  044070          100$:   JSR     PC,GETSTS ;SETUP FOR STATUS FETCH
1588 031514  004737  044766                  JSR     PC,TIMOUT ;WAIT FOR COMPLETION
1589
1590 031520  004737  044154                  JSR     PC,GET    ;GO READ REGISTER(S) WITH GET SUBROUTINE
     031524  000404                          BR      110$      ;GO TO 110$ IF NO ERROR
     031526  000240                          NOP               ;RETURN HERE IF ERROR
     031530  104000                          EMT               ;ERROR # DEFINED BY GET SUBROUTINE
     031532  000137  031660                  JMP     160$      ;GO TO 160$ IF ERROR
1591 031536                          110$:
1592 031536  004737  045152                  JSR     PC,PRIERR ;GO CHECK FOR PRIMARY ERRORS
     031542  000405                          BR      120$      ;GO TO 120$ IF NO ERROR
     031544  000240                          NOP               ;RETURN HERE IF ERROR
     031546  104000                          EMT               ;ERROR # DEFINED BY PRIERR SUBROUTINE
     031550  004736                          JSR     PC,@(SP)+ ;GO BACK FOR MORE ERROR CHECKS
     031552  000137  031660                  JMP     160$      ;GO TO 160$ IF ERROR
1593 031556                          120$:
1594 031556  004737  052270                  JSR     PC,SEKSTS ;GO VERIFY RESULTS OF SEEK OPERATION
     031562  000405                          BR      130$      ;GO TO 130$ IF NO ERROR
     031564  000240                          NOP               ;RETURN HERE IF ERROR
     031566  104000                          EMT               ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     031570  004736                          JSR     PC,@(SP)+ ;GO BACK FOR MORE ERROR CHECKS
     031572  000137  031660                  JMP     160$      ;GO TO 160$ IF ERROR
1595 031576                          130$:
1596 031576  004737  051644                  JSR     PC,CMPERRSTS ;CHECK ANY ERRORS NOT MASKED
     031602  115760                          .WORD   NDTMSK    ;MASK FOR RMER1
     031604  000010                          .WORD   DPE       ;MASK FOR RMER2
     031606  000405                          BR      140$      ;GO TO 140$ IF NO ERROR
     031610  000240                          NOP               ;RETURN HERE IF ERROR
     031612  104000                          EMT               ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     031614  004736                          JSR     PC,@(SP)+ ;GO BACK FOR MORE ERROR CHECKS
     031616  000137  031660                  JMP     160$      ;GO TO 160$ IF ERROR
1597 031622                          140$:
```

```
1598 031622  004737  046004            JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     031626  000405                    BR      150$            ;GO TO 150$ IF NO ERROR
     031630  000240                    NOP                     ;RETURN HERE IF ERROR
     031632  104000                    EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     031634  004736                    JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     031636  000137  031660            JMP     160$            ;GO TO 160$ IF ERROR
1599 031642                        150$:
1600 031642  005737  001444            TST     RMDCO           ;FIRST TIME THROUGH??
1601 031646  001004                    BNE     160$            ;NO!!
1602 031650  005237  001444            INC     RMDCO           ;YES - SEEK TO ADJACENT CYLINDER FORWARD
1603 031654  000137  031472            JMP     90$
1604 031660                        160$:
1605
1606                                   ;;*******************************************************************
                                       ;*TEST 51       SEEK ADJACENT REVERSE
                                       ;;*******************************************************************
     031660                        TST51:
     031660  000004                    SCOPE                   ;SCOPE CALL
     031662  000240                    NOP                     ;START OF TEST
     031664  012706  001100            MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     031670  013700  001276            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     031674  013701  001464            MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     031700  012737  000001  001206    MOV     #1,$TIMES       ;;DO 1 ITERATION
     031706  012737  000051  001226    MOV     #51,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1607
1608 031714                        10$:
1609 031714  004737  043150            JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     031720  054130                    .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                               ;CLEAR CONTROLLER & SELECT DEVICE
                                                               ;VERIFY CONTROLLER CLEAR OPERATION
                                                               ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                               ;VERIFY PACK ACKNOWLEDGE
                                                               ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                               ;VERIFY RECALIBRATION
     031722  000404                    BR      80$             ;GO TO 80$ IF NO ERROR
     031724  000240                    NOP                     ;RETURN HERE IF ERROR
     031726  104000                    EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     031730  000137  032172            JMP     160$            ;GO TO 160$ IF ERROR
1610 031734                        80$:
1611 031734  012737  000001  001444    MOV     #1,RMDCO        ;CYLINDER = 1
1612 031742  012737  000000  001416    MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
1613 031750  012737  000005  001410    MOV     #SEEK!GO,RMCS1O ;FUNCTION CODE FOR SEEK
1614 031756  012702  001543            MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER INDEX
1615 031762  112722  000006            MOVB    #RMDA,(R2)+     ;WRITE REGISTER INDEX TABLE
1616 031766  112722  000034            MOVB    #RMDC,(R2)+
1617 031772  112722  000000            MOVB    #RMCS1,(R2)+
1618 031776  112722  000200            MOVB    #200,(R2)+      ;TERMINATE TABLE
1619 032002                        90$:
1620 032002  004737  044424            JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032006  000404                    BR      100$            ;GO TO 100$ IF NO ERROR
     032010  000240                    NOP                     ;RETURN HERE IF ERROR
     032012  104000                    EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032014  000137  032172            JMP     160$            ;GO TO 160$ IF ERROR
1621 032020  004737  044070        100$:JSR    PC,GETSTS       ;SETUP TO READ ALL REGISTERS
1622 032024  004737  044766            JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
1623
1624 032030  004737  044154            JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
```

```
        032034  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
        032036  000240                          NOP                     ;RETURN HERE IF ERROR
        032040  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        032042  000137  032172                  JMP     160$            ;GO TO 160$ IF ERROR
  1625  032046                          110$:
  1626  032046  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        032052  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        032054  000240                          NOP                     ;RETURN HERE IF ERROR
        032056  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        032060  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032062  000137  032172                  JMP     160$            ;GO TO 160$ IF ERROR
  1627  032066                          120$:
  1628  032066  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        032072  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
        032074  000240                          NOP                     ;RETURN HERE IF ERROR
        032076  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        032100  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032102  000137  032172                  JMP     160$            ;GO TO 160$ IF ERROR
  1629  032106                          130$:
  1630  032106  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        032112  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
        032114  000010                          .WORD   DPE             ;MASK FOR RMER2
        032116  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        032120  000240                          NOP                     ;RETURN HERE IF ERROR
        032122  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        032124  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032126  000137  032172                  JMP     160$            ;GO TO 160$ IF ERROR
  1631  032132                          140$:
  1632  032132  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        032136  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        032140  000240                          NOP                     ;RETURN HERE IF ERROR
        032142  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        032144  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032146  000137  032172                  JMP     160$            ;GO TO 160$ IF ERROR
  1633  032152                          150$:
  1634  032152  022737  000001  001444          CMP     #1,RMDCO        ;IS THIS THE FIRST SEEK??
  1635  032160  001004                          BNE     160$            ;NO!!
  1636  032162  005337  001444                  DEC     RMDCO           ;YES - SEEK TO ADJACENT CYLINDER
  1637  032166  000137  032002                  JMP     90$             ;GO SEEK
  1638  032172                          160$:
  1639
  1640                                  ;;*******************************************************************
                                        ;*TEST 52        SEEK INVALID SECTOR
                                        ;;*******************************************************************
        032172                          TST52:
        032172  000004                          SCOPE                   ;SCOPE CALL
        032174  000240                          NOP                     ;START OF TEST
        032176  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        032202  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        032206  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        032212  012737  000052  001226          MOV     #52,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  1641
  1642  032220                          10$:
  1643  032220  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        032224  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
```

```
                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                        ;VERIFY PACK ACKNOWLEDGE
                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                        ;VERIFY RECALIBRATION
     032226  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
     032230  000240                          NOP                     ;RETURN HERE IF ERROR
     032232  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     032234  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1644 032240                          80$:
1645 032240  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;SEEK COMMAND
1646 032246  012737  000000  001442          MOV     #0,RMOFO        ;FORMAT 18 BIT
1647 032254  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
1648 032262  012737  000036  001416          MOV     #30.,RMDAO      ;INVALID SECTOR = 30., TRACK = 0
1649 032270  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
1650 032274  112722  000032                  MOVB    #RMOF,(R2)+     ;WRITE REGISTER INDEX TABLE
1651 032300  112722  000006                  MOVB    #RMDA,(R2)+
1652 032304  112722  000034                  MOVB    #RMDC,(R2)+
1653 032310  112722  000000                  MOVB    #RMCS1,(R2)+
1654 032314  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
1655 032320  004737  044070                  JSR     PC,GETSTS       ;SETUP INPUT REGISTER INDEX
1656 032324                          90$:
1657 032324  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
     032330  000404                          BR      95$             ;GO TO 95$ IF NO ERROR
     032332  000240                          NOP                     ;RETURN HERE IF ERROR
     032334  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
     032336  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1658 032342                          95$:
1659 032342  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032346  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
     032350  000240                          NOP                     ;RETURN HERE IF ERROR
     032352  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032354  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1660 032360                          100$:
1661 032360  004737  053646                  JSR     PC,CLRSTS       ;GO VERIFY CONTROLLER CLEAR OPERATION
     032364  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
     032366  000240                          NOP                     ;RETURN HERE IF ERROR
     032370  104000                          EMT                     ;ERROR # DEFINED BY CLRSTS SUBROUTINE
     032372  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     032374  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1662 032400                          110$:
1663 032400  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     032404  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
     032406  000240                          NOP                     ;RETURN HERE IF ERROR
     032410  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     032412  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1664 032416                          120$:
1665 032416  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
1666
1667 032422  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     032426  000404                          BR      125$            ;GO TO 125$ IF NO ERROR
     032430  000240                          NOP                     ;RETURN HERE IF ERROR
     032432  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     032434  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1668 032440                          125$:
1669 032440  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     032444  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
     032446  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
        032450  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        032452  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032454  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1670    032460                          130$:
1671    032460  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        032464  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        032466  000240                          NOP                     ;RETURN HERE IF ERROR
        032470  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        032472  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032474  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1672    032500                          140$:
1673    032500                          150$:
1674    032500  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        032504  117760                          .WORD   IAE!NDTMSK              ;MASK FOR RMER1
        032506  000010                          .WORD   DPE             ;MASK FOR RMER2
        032510  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        032512  000240                          NOP                     ;RETURN HERE IF ERROR
        032514  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        032516  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR C 'ECKS
        032520  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1675    032524                          160$:
1676    032524  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        032530  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
        032532  000240                          NOP                     ;RETURN HERE IF ERROR
        032534  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        032536  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        032540  000137  032610                  JMP     180$            ;GO TO 180$ IF ERROR
1677    032544                          170$:
1678    032544  005237  001416                  INC     RMDAO           ;INCREMENT SECTOR ADDRESS
1679    032550  022737  000037  001416          CMP     #31.,RMDAO      ;DONE??
1680    032556  103262                          BHIS    90$             ;NO - TEST NEXT SECTOR
1681
1682    032560  032737  010000  001442          BIT     #FMT16,RMOFO    ;WAS TEST DONE FOR 16 BIT MODE YET ?
1683    032566  001010                          BNE     180$            ;BR IF YES
1684    032570  012737  010000  001442          MOV     #FMT16,RMOFO    ;SET 16 BIT FORMAT MODE
1685    032576  012737  000037  001416          MOV     #31.,RMDAO      ;SET INVALID SECTOR 31.
1686    032604  000137  032324                  JMP     90$             ;NO - TEST NEXT SECTOR
1687    032610                          180$:
1688
1689                                    ;;*****************************************************************
                                        ;*TEST 53        SEEK INVALID TRACK
                                        ;;*****************************************************************
        032610                          TST53:
        032610  000004                          SCOPE                   ;SCOPE CALL
        032612  000240                          NOP                     ;START OF TEST
        032614  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        032620  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        032624  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        032630  012737  000053  001226          MOV     #53,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1690
1691    032636  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        032642  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
```

```
                                                        ;VERIFY RECALIBRATION
        032644  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        032646  000240                          NOP                     ;RETURN HERE IF ERROR
        032650  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        032652  000137  033236                  JMP     200$            ;GO TO 200$ IF ERROR
1692    032656                          80$:
1693    032656  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;SEEK COMMAND
1694    032664  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
1695    032672  013737  001332  001416          MOV     LSTRK,RMDA0     ;LAST TRACK, SECTOR = 0
1696    032700  105237  001417                  INCB    RMDA0+1         ;SETUP FIRST INVALID TRACK
1697    032704  012737  000000  001442          MOV     #0,RMOF0        ;18 BIT FORMAT
1698    032712  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
1699    032716  112722  000032                  MOVB    #RMOF,(R2)+     ;WRITE REGISTER INDEX
1700    032722  112722  000034                  MOVB    #RMDC,(R2)+     ;TABLE FOR OUTPUT
1701    032726  112722  000006                  MOVB    #RMDA,(R2)+
1702    032732  112722  000000                  MOVB    #RMCS1,(R2)+
1703    032736  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
1704    032742  004737  044070                  JSR     PC,GETSTS       ;SETUP INPUT TABLE FOR STATUS
1705    032746                          90$:
1706    032746  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
        032752  000404                          BR      95$             ;GO TO 95$ IF NO ERROR
        032754  000240                          NOP                     ;RETURN HERE IF ERROR
        032756  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
        032760  000137  033236                  JMP     200$            ;GO TO 200$ IF ERROR
1707    032764                          95$:
1708    032764  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        032770  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
        032772  000240                          NOP                     ;RETURN HERE IF ERROR
        032774  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        032776  000137  033236                  JMP     200$            ;GO TO 200$ IF ERROR
1709    033002                          100$:
1710    033002  004737  053646                  JSR     PC,CLRSTS       ;GO VERIFY CONTROLLER CLEAR OPERATION
        033006  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        033010  000240                          NOP                     ;RETURN HERE IF ERROR
        033012  104000                          EMT                     ;ERROR # DEFINED BY CLRSTS SUBROUTINE
        033014  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        033016  000137  033236                  JMP     200$            ;GO TO 200$ IF ERROR
1711    033022                          110$:
1712    033022  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        033026  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
        033030  000240                          NOP                     ;RETURN HERE IF ERROR
        033032  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        033034  000137  033236                  JMP     200$            ;GO TO 200$ IF ERROR
1713    033040                          120$:
1714    033040  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
1715
1716    033044  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        033050  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
        033052  000240                          NOP                     ;RETURN HERE IF ERROR
        033054  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        033056  000137  033236                  JMP     200$            ;GO TO 200$ IF ERROR
1717    033062                          130$:
1718    033062  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        033066  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        033070  000240                          NOP                     ;RETURN HERE IF ERROR
        033072  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        033074  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
```

```
     033076  000137  033236                    JMP     200$            ;GO TO 200$ IF ERROR
1719 033102                             140$:
1720 033102  004737  052270                    JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     033106  000405                            BR      150$            ;GO TO 150$ IF NO ERROR
     033110  000240                            NOP                     ;RETURN HERE IF ERROR
     033112  104000                            EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     033114  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033116  000137  033236                    JMP     200$            ;GO TO 200$ IF ERROR
1721 033122                             150$:
1722 033122                             160$:
1723 033122  004737  051644                    JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     033126  117760                            .WORD   IAE!NDTMSK              ;MASK FOR RMER1
     033130  000010                            .WORD   DPE             ;MASK FOR RMER2
     033132  000405                            BR      170$            ;GO TO 170$ IF NO ERROR
     033134  000240                            NOP                     ;RETURN HERE IF ERROR
     033136  104000                            EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     033140  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033142  000137  033236                    JMP     200$            ;GO TO 200$ IF ERROR
1724 033146                             170$:
1725 033146  004737  046004                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     033152  000405                            BR      180$            ;GO TO 180$ IF NO ERROR
     033154  000240                            NOP                     ;RETURN HERE IF ERROR
     033156  104000                            EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     033160  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     033162  000137  033236                    JMP     200$            ;GO TO 200$ IF ERROR
1726 033166                             180$:
1727 033166  105237  001417                    INCB    RMDAO+1         ;INCREMENT TRACK ADDRESS
1728 033172  123727  001417  000200            CMPB    RMDAO+1,#128.   ;DONE??
1729 033200  101414                            BLOS    195$            ;NO-DO NEXT TRACK
1730
1731 033202  032737  010000  001442            BIT     #FMT16,RMOFO    ;WAS TEST DONE FOR 16 BIT??
1732 033210  001012                            BNE     200$            ;YES!!
1733 033212  012737  010000  001442            MOV     #FMT16,RMOFO    ;SET FORMAT 16
1734 033220  013737  001332  001416            MOV     LSTRK,RMDAO     ;LAST TRACK, SECTOR = 0
1735 033226  105237  001417                    INCB    RMDAO+1         ;SETUP FIRST INVALID TRACK ADDRESS
1736 033232  000137  032746            195$:   JMP     90$             ;DO TEST FOR 16 BIT FORMAT
1737
1738 033236                             200$:
1739
1740                                    ;;*****************************************************************
                                        ;*TEST 54        SEEK INVALID CYLINDER
                                        ;;*****************************************************************
     033236                             TST54:
     033236  000004                            SCOPE                   ;SCOPE CALL
     033240  000240                            NOP                     ;START OF TEST
     033242  012706  001100                    MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     033246  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     033252  013701  001464                    MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     033256  012737  000054  001226            MOV     #54,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1741
1742 033264  004737  043150                    JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     033270  054130                            .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
```

```
                033272  000404                          BR      80$             ;VERIFY RECALIBRATION
                033274  000240                          NOP                     ;GO TO 80$ IF NO ERROR
                033276  104000                          EMT                     ;RETURN HERE IF ERROR
                033300  000137  033654                  JMP     200$            ;ERROR # DEFINED BY TSTPRP SUBROUTINE
1743            033304                          80$:                            ;GO TO 200$ IF ERROR
1744            033304  012737  000005  001410          MOV     #SEEK.GO,RMCS10 ;SEEK FUNCTION CODE
1745            033312  012737  001467  001444          MOV     #823.,RMDC0     ;START AT FIRST INVALID CYLINDER
1746            033320  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
1747            033326  012737  000000  001442          MOV     #0,RMOF0        ;18 BIT FORMAT
1748            033334  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO INDEX TABLE
1749            033340  112722  000006                  MOVB    #RMDA,(R2)+     ;WRITE REGISTER TABLE
1750            033344  112722  000034                  MOVB    #RMDC,(R2)+
1751            033350  112722  000032                  MOVB    #RMOF,(R2)+
1752            033354  112722  000000                  MOVB    #RMCS1,(R2)+
1753            033360  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
1754            033364  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS
1755            033370                          90$:
1756            033370  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
                033374  000404                          BR      95$             ;GO TO 95$ IF NO ERROR
                033376  000240                          NOP                     ;RETURN HERE IF ERROR
                033400  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
                033402  000137  033654                  JMP     200$            ;GO TO 200$ IF ERROR
1757            033406                          95$:
1758            033406  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                033412  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
                033414  000240                          NOP                     ;RETURN HERE IF ERROR
                033416  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                033420  000137  033654                  JMP     200$            ;GO TO 200$ IF ERROR
1759            033424                          100$:
1760            033424  004737  053646                  JSR     PC,CLRSTS       ;GO VERIFY CONTROLLER CLEAR OPERATION
                033430  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
                033432  000240                          NOP                     ;RETURN HERE IF ERROR
                033434  104000                          EMT                     ;ERROR # DEFINED BY CLRSTS SUBROUTINE
                033436  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                033440  000137  033654                  JMP     200$            ;GO TO 200$ IF ERROR
1761            033444                          110$:
1762            033444  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                033450  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
                033452  000240                          NOP                     ;RETURN HERE IF ERROR
                033454  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                033456  000137  033654                  JMP     200$            ;GO TO 200$ IF ERROR
1763            033462                          120$:
1764            033462  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
1765
1766            033466  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                033472  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
                033474  000240                          NOP                     ;RETURN HERE IF ERROR
                033476  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                033500  000137  033654                  JMP     200$            ;GO TO 200$ IF ERROR
1767            033504                          130$:
1768            033504  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                033510  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
                033512  000240                          NOP                     ;RETURN HERE IF ERROR
                033514  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                033516  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                033520  000137  033654                  JMP     200$            ;GO TO 200$ IF ERROR
```

```
1769 033524
1770 033524  004737  052270          140$:
     033530  000405                      JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     033532  000240                      BR      150$            ;GO TO 150$ IF NO ERROR
     033534  104000                      NOP                     ;RETURN HERE IF ERROR
     033536  004736                      EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     033540  000137  033654              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1771 033544                              JMP     200$            ;GO TO 200$ IF ERROR
1772 033544  004737  051644          150$:
     033550  117760                      JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     033552  000010                      .WORD   IAE!NDTMSK              ;MASK FOR RMER1
     033554  000405                      .WORD   DPE             ;MASK FOR RMER2
     033556  000240                      BR      170$            ;GO TO 170$ IF NO ERROR
     033560  104000                      NOP                     ;RETURN HERE IF ERROR
     033562  004736                      EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     033564  000137  033654              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1773 033570                              JMP     200$            ;GO TO 200$ IF ERROR
1774 033570  004737  046004          170$:
     033574  000405                      JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     033576  000240                      BR      180$            ;GO TO 180$ IF NO ERROR
     033600  104000                      NOP                     ;RETURN HERE IF ERROR
     033602  004736                      EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     033604  000137  033654              JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
1775 033610                              JMP     200$            ;GO TO 200$ IF ERROR
1776 033610  005237  001444          180$:
1777 033614  023727  001444  002000      INC     RMDCO           ;INCREMENT CYLINDER ADDRESS
1778 033622  002412                      CMP     RMDCO,#1024.    ;DONE??
1779                                     BLT     195$            ;NO-DO NEXT CYLINDER
1780 033624  032737  010000  001442      BIT     #FMT16,RMOFO    ;16 BIT FORMAT TESTED??
1781 033632  001010                      BNE     200$            ;YES .!
1782 033634  012737  010000  001442      MOV     #FMT16,RMOFO    ;SETUP FOR 16 BIT TEST
1783 033642  012737  001467  001444      MOV     #823.,RMDCO     ;START AT FIRST INVALID CYLINDER
1784 033650  000137  033370      195$:   JMP     90$
1785
1786 033654                      200$:
1787
1788                                     ;;*****************************************************************
                                         ;*TEST 55        IVC SEEK TEST
                                         ;;*****************************************************************
     033654                      TST55:
     033654  000004                      SCOPE                   ;SCOPE CALL
     033656  000240                      NOP                     ;START OF TEST
     033660  012706  001100              MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     033664  013700  001276              MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     033670  013701  001464              MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     033674  012737  000055  001226      MOV     #55,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1789
1790 033702  004737  053530              JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
     033706  000404                      BR      10$             ;GO TO 10$ IF NO ERROR
     033710  000240                      NOP                     ;RETURN HERE IF ERROR
     033712  104000                      EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
     033714  000137  034214              JMP     90$             ;GO TO 90$ IF ERROR
1791 033720                      10$:
1792 033720  112737  000024  001543      MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
     033726  112737  000200  001544      MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
     033734  012737  000001  001434      MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER - DMD
     033742  004737  044424              JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
```

```
        033746  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
        033750  000240                          NOP                     ;RETURN HERE IF ERROR
        033752  104000                          EMT                     ;ERROR DEFINED BY PU" SUBROUTINE
        033754  000137  034214                  JMP     90$             ;GO TO 90$ IF ERROR
  1793  033760                          20$:
  1794  033760  012737  000005  001410          MOV     #SEEK.GO,RMCS10
  1795  033766  012737  000000  001434          MOV     #0,RMMR10
  1796  033774  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
  1797  034002  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
  1798  034010  012702  001543                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX
  1799  034014  112722  000024                  MOVB    #RMMR1,(R2)+    ;TABLE
  1800  034020  112722  000006                  MOVB    #RMDA,(R2)+
  1801  034024  112722  000034                  MOVB    #RMDC,(R2)+
  1802  034030  112722  000000                  MOVB    #RMCS1,(R2)+
  1803  034034  112722  000200                  MOVB    #200,(R2)+
  1804  034040  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS
  1805  034044  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        034050  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        034052  000240                          NOP                     ;RETURN HERE IF ERROR
        034054  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        034056  000137  034214                  JMP     90$             ;GO TO 90$ IF ERROR
  1806  034062                          30$:
  1807  034062  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        034066  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
        034070  000240                          NOP                     ;RETURN HERE IF ERROR
        034072  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        034074  000137  034214                  JMP     90$             ;GO TO 90$ IF ERROR
  1808  034100                          40$:
  1809  034100  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        034104  000405                          BR      50$             ;GO TO 50$ IF NO ERROR
        034106  000240                          NOP                     ;RETURN HERE IF ERROR
        034110  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        034112  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        034114  000137  034214                  JMP     90$             ;GO TO 90$ IF ERROR
  1810  034120  032737  010000  001376  50$:    BIT     #IVC,RMER2I     ;DID INVALID COMMAND SET??
  1811  034126  001012                          BNE     60$             ;YES!!
  1812  034130  012737  010000  001140          MOV     #IVC,$GDDAT     ;GOOD DATA FOR TYPEOUT
  1813  034136  013737  001376  001142          MOV     RMER2I,$BDDAT   ;BAD DATA FOR TYPEOUT
  1814  034144  042737  167777  001142          BIC     #^CIVC,$BDDAT
  1815  034152  104064                          EMT     64
  1816  034154                          60$:
  1817  034154  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        034160  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
        034162  010010                          .WORD   IVC!DPE         ;MASK FOR RMER2
        034164  000405                          BR      80$             ;GO TO 80$ IF NO ERROR
        034166  000240                          NOP                     ;RETURN HERE IF ERROR
        034170  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        034172  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        034174  000137  034214                  JMP     90$             ;GO TO 90$ IF ERROR
  1818  034200                          80$:
  1819  034200  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        034204  000403                          BR      90$             ;GO TO 90$ IF NO ERROR
        034206  000240                          NOP                     ;RETURN HERE IF ERROR
        034210  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        034212  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
  1820  034214                          90$:
  1821
```

```
1822                                    ;;*****************************************************************
                                        ;*TEST 56       ABORT SEEK TEST
                                        ;;*****************************************************************
        034214                          TST56:
        034214  000004                          SCOPE                   ;SCOPE CALL
        034216  000240                          NOP                     ;START OF TEST
        034220  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        034224  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        034230  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        034234  012737  000056  001226          MOV     #56,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1823
1824    034242  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        034246  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        034250  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        034252  000240                          NOP                     ;RETURN HERE IF ERROR
        034254  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        034256  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
1825    034262                          80$:
1826    034262  012737  040000  001424          MOV     #UNS,RMER10     ;SET UNSAFE ERROR
1827    034270  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
1828    034276  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
1829    034304  012737  000005  001410          MOV     #SEEK!GO,RMCS10 ;SEEK COMMAND
1830    034312  012702  001543                  MOV     #PUTINX,R2      ;SETUP REGISTER INDEX TABLE
1831    034316  112722  000006                  MOVB    #RMDA,(R2)+     ;AND OUTPUT BUFFER
1832    034322  112722  000034                  MOVB    #RMDC,(R2)+
1833    034326  112722  000014                  MOVB    #RMER1,(R2)+
1834    034332  112722  000000                  MOVB    #RMCS1,(R2)+
1835    034336  112722  000200                  MOVB    #200,(R2)+
1836    034342  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
1837    034346  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        034352  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        034354  000240                          NOP                     ;RETURN HERE IF ERROR
        034356  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        034360  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
1838    034364                          90$:
1839    034364  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        034370  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
        034372  000240                          NOP                     ;RETURN HERE IF ERROR
        034374  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        034376  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
1840    034402  032737  020000  001346  100$:   BIT     #PIP,RMDS1      ;DID DRIVE START SEEK??
1841    034410  001411                          BEQ     110$            ;NO!!
1842    034412  013737  001346  001142          MOV     RMDS1,$BDDAT    ;BAD DATA FOR TYPEOUT
1843    034420  042737  157777  001142          BIC     #^CPIP,$BDDAT
1844    034426  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
1845    034432  104065                          EMT     65
1846    034434  004737  044766          110$:   JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
1847
1848    034440  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        034444  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
        034446  000240                          NOP                     ;RETURN HERE IF ERROR
```

```
                034450  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                034452  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
     1849       034456                          120$:
     1850       034456  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                034462  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
                034464  000240                          NOP                     ;RETURN HERE IF ERROR
                034466  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                034470  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                034472  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
     1851       034476                          130$:
     1852       034476  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
                034502  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
                034504  000240                          NOP                     ;RETURN HERE IF ERROR
                034506  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
                034510  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                034512  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
     1853       034516                          140$:
     1854       034516  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
                034522  155760                          .WORD   NDTMSK!UNS              ;MASK FOR RMER1
                034524  000010                          .WORD   DPE             ;MASK FOR RMER2
                034526  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                034530  000240                          NOP                     ;RETURN HERE IF ERROR
                034532  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
                034534  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                034536  000137  034556                  JMP     160$            ;GO TO 160$ IF ERROR
     1855       034542                          150$:
     1856       034542  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                034546  000403                          BR      160$            ;GO TO 160$ IF NO ERROR
                034550  000240                          NOP                     ;RETURN HERE IF ERROR
                034552  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                034554  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     1857       034556                          160$:
     1858
     1859                                       ;;*****************************************************************
                                                ;*TEST 57        SEEK AT OFFSET
                                                ;;*****************************************************************
                034556                          TST57:
                034556  000004                          SCOPE                   ;SCOPE CALL
                034560  000240                          NOP                     ;START OF TEST
                034562  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                034566  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                034572  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                034576  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
                034604  012737  000057  001226          MOV     #57,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     1860
     1861       034612                          10$:
     1862       034612  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                034616  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                034620  000404                          BR      20$             ;GO TO 20$ IF NO ERROR
                034622  000240                          NOP                     ;RETURN HERE IF ERROR
                034624  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
```

```
        034626  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
1863    034632                          20$:
1864    034632  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
1865    034640  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
1866    034646  012737  000015  001410          MOV     #OFFSET!GO,RMCS1O       ;LOAD OFFSET COMMAND IN BUFFER
1867    034654  012702  001543                  MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
1868    034660  112722  000034                  MOVB    #RMDC,(R2)+     ;WRITE REGISTER INDEX TABLE
1869    034664  112722  000006                  MOVB    #RMDA,(R2)+
1870    034670  112722  000000                  MOVB    #RMCS1,(R2)+
1871    034674  112722  000200                  MOVB    #200,(R2)+      ;WRITE TERMINATOR
1872    034700  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        034704  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
        034706  000240                          NOP                     ;RETURN HERE IF ERROR
        034710  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        034712  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
1873    034716  004737  044070          30$:    JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
1874    034722  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR OFFSET TO COMPLETE
1875    034726                          40$:
1876    034726  012737  000005  001410          MOV     #SEEK!GO,RMCS1O ;LOAD SEEK COMMAND
1877    034734  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;LOAD REGISTER INDEX TABLE
1878    034742  112737  000200  001544          MOVB    #200,PUTINX+1
1879    034750  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        034754  000404                          BR      50$             ;GO TO 50$ IF NO ERROR
        034756  000240                          NOP                     ;RETURN HERE IF ERROR
        034760  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        034762  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
1880    034766                          50$:
1881
1882                                    ;WAIT FOR COMMAND TO COMPLETE
        034766  004737  044766                  JSR     PC,TIMOUT       ;GO TO TIMOUT SUBROUTINE
1883
1884    034772  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        034776  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
        035000  000240                          NOP                     ;RETURN HERE IF ERROR
        035002  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        035004  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
1885    035010                          100$:
1886    035010  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        035014  000405                          BR      110$            ;GO TO 110$ IF NO ERROR
        035016  000240                          NOP                     ;RETURN HERE IF ERROR
        035020  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        035022  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035024  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
1887    035030                          110$:
1888    035030  004737  052270                  JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
        035034  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        035036  000240                          NOP                     ;RETURN HERE IF ERROR
        035040  104000                          EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
        035042  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        035044  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
1889    035050                          120$:
1890    035050  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        035054  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
        035056  000010                          .WORD   DPE             ;MASK FOR RMER2
        035060  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
        035062  000240                          NOP                     ;RETURN HERE IF ERROR
        035064  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
```

```
                035066  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                035070  000137  035110                  JMP     140$            ;GO TO 140$ IF ERROR
     1891       035074                          130$:
     1892       035074  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                035100  000403                          BR      140$            ;GO TO 140$ IF NO ERROR
                035102  000240                          NOP                     ;RETURN HERE IF ERROR
                035104  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                035106  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     1893       035110                          140$:
     1894
     1895                                       ;;*****************************************************************
                                                ;*TEST 60        LOOK AHEAD TEST
                                                ;;*****************************************************************
                035110                          TST60:
                035110  000004                          SCOPE                   ;SCOPE CALL
                035112  000240                          NOP                     ;START OF TEST
                035114  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                035120  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                035124  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                035130  012737  000060  001226          MOV     #60,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
     1896
     1897       035136  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
                035142  000404                          BR      10$             ;GO TO 10$ IF NO ERROR
                035144  000240                          NOP                     ;RETURN HERE IF ERROR
                035146  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
                035150  000137  035536                  JMP     160$            ;GO TO 160$ IF ERROR
     1898       035154                          10$:
     1899       035154  012703  000037                  MOV     #31.,R3         ;R3 = SAMPLE COUNT FOR 18 BIT MODE
     1900       035160  012737  000000  001442          MOV     #0,RMOFO        ;START WITH 18 BIT MODE
     1901       035166  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
     1902       035174  012737  000000  001416          MOV     #0,RMDAO        ;SEARCH TRACK = 0, SECTOR - 0
     1903       035202  012737  000031  001410          MOV     #SEARCH!GO,RMCS10       ;RECALIBRATE COMMAND
     1904       035210  012702  001543                  MOV     #PUTINX,R2
     1905       035214  112722  000006                  MOVB    #RMDA,(R2)+
     1906       035220  112722  000034                  MOVB    #RMDC,(R2)+
     1907       035224  112722  000032                  MOVB    #RMOF,(R2)+
     1908       035230  112722  000000                  MOVB    #RMCS1,(R2)+
     1909       035234  112722  000200                  MOVB    #200,(R2)+
     1910       035240  004737  044070                  JSR     PC,GETSTS
     1911       035244                          20$:
     1912       035244  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                035250  000404                          BR      30$             ;GO TO 30$ IF NO ERROR
                035252  000240                          NOP                     ;RETURN HERE IF ERROR
                035254  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                035256  000137  035526                  JMP     150$            ;GO TO 150$ IF ERROR
     1913       035262  004737  044766          30$:    JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
     1914
     1915       035266  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
                035272  000404                          BR      40$             ;GO TO 40$ IF NO ERROR
                035274  000240                          NOP                     ;RETURN HERE IF ERROR
                035276  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
                035300  000137  035526                  JMP     150$            ;GO TO 150$ IF ERROR
     1916       035304                          40$:
     1917       035304  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                035310  000405                          BR      50$             ;GO TO 50$ IF NO ERROR
                035312  000240                          NOP                     ;RETURN HERE IF ERROR
                035314  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
```

```
035316  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
035320  000137  035526                   JMP     150$            ;GO TO 150$ IF ERROR
1918 035324                     50$:
1919 035324  012704  103640     60$:    MOV     #BUFFER,R4      ;R4 = STORAGE ADDRESS
1920 035330  012705  177777             MOV     #-1,R5          ;R5 = GROSS TIMER
1921 035334  016014  000020     70$:    MOV     RMLA(R0),(R4)   ;STORE RMLA SAMPLE
1922 035340  016002  000020             MOV     RMLA(R0),R2     ;GET ANOTHER LOOK
1923 035344  020214                     CMP     R2,(R4)         ;ARE SAMPLES SAME??
1924 035346  001403                     BEQ     80$             ;YES!!
1925 035350  005305                     DEC     R5              ;TIMEOUT??
1926 035352  001370                     BNE     70$             ;NO - TRY AGAIN
1927 035354  000411                     BR      100$            ;PROGRAM TIMEOUT
1928 035356  062704  000002     80$:    ADD     #2,R4           ;ADVANCE STORAGE ADDRESS
1929 035362  005303                     DEC     R3              ;ALL SAMPLES TAKEN??
1930 035364  001410                     BEQ     110$            ;YES!!
1931 035366  026002  000020     90$:    CMP     RMLA(R0),R2     ;WAIT FOR CHANGE
1932 035372  001360                     BNE     70$             ;YES!!
1933 035374  005305                     DEC     R5              ;TIMEOUT??
1934 035376  001373                     BNE     90$             ;NO
1935 035400                     100$:
     035400  104352                     EMT     352
1936 035402  000137  035536             JMP     160$
1937
1938                            ;SET 18 BIT SAMPLE PARAMETERS
1939 035406  012702  003500     110$:   MOV     #3500,R2        ;R2 = MAXIMUM SAMPLE
1940 035412  012703  000036             MOV     #30.,R3         ;R3 = NUMBER OF COMPARES
1941 035416  012704  103640             MOV     #BUFFER,R4      ;R4 = BUFFER ADDRESS
1942 035422  032737  010000  001442     BIT     #FMT16,RMOFO    ;IS SAMPLE FOR 18 BIT MODE ?
1943 035430  001404                     BEQ     120$            ;BR IF YES
1944
1945                            ;SET 16 BIT SAMPLE PARAMETERS
1946 035432  012702  003700             MOV     #3700,R2        ;R2 = MAXIMUM SAMPLE
1947 035436  012703  000040             MOV     #32.,R3         ;R3 = NUMBER OF COMPARES
1948 035442  012405             120$:   MOV     (R4)+,R5        ;GET A SAMPLE AND INCREMENT
1949 035444  062705  000100             ADD     #100,R5         ;FOR EXPECTED VALUE OF NEXT
1950 035450  020205                     CMP     R2,R5           ;SHOULD NEXT BE 0 ?
1951 035452  103001                     BHIS    130$            ;NO!!
1952 035454  005005                     CLR     R5              ;YES - CHANGE EXPECTED
1953 035456  020514             130$:   CMP     R5,(R4)         ;IS NEXT SAMPLE CORRECT??
1954 035460  001406                     BEQ     140$            ;YES!!
1955 035462  010537  001140             MOV     R5,$GDDAT       ;EXPECTED VALUE
1956 035466  011437  001142             MOV     (R4),$BDDAT     ;RECEIVED VALUE
1957 035472  104353                     EMT     353
1958 035474  000414                     BR      150$
1959 035476  005303             140$:   DEC     R3              ;ALL SAMPLES CHECKED??
1960 035500  001360                     BNE     120$            ;NO - TEST NEXT SAMPLE
1961
1962 035502  032737  010000  001442     BIT     #FMT16,RMOFO    ;TEST DONE FOR 16 BIT MODE ?
1963 035510  001006                     BNE     150$            ;BR IF YES
1964 035512  012737  010000  001442     MOV     #FMT16,RMOFO    ;SET 16 BIT MODE AND
1965 035520  012703  000041             MOV     #33.,R3         ;SET SAMPLE COUNT.
1966 035524  000647                     BR      20$             ;TEST AGAIN
1967
1968 035526                     150$:
     035526  012637  000006             MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
1969 035532  012637  000004             MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
1970 035536                     160$:
```

```
1971
1972                                    ;;*****************************************************************
                                        ;*TEST 61        SEARCH ON CYLINDER
                                        ;;*****************************************************************
                                        TST61:
     035536                                     SCOPE                       ;SCOPE CALL
     035536  000004                             NOP                         ;START OF TEST
     035540  000240                             MOV     #STACK,SP           ;INITIALIZE STACK POINTER
     035542  012706  001100                     MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS
     035546  013700  001276                     MOV     TSTQUE,R1           ;(R1) = DEVICE BEING TESTED
     035552  013701  001464                     MOV     #61,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
     035556  012737  000061  001226
1973
1974 035564  004737  043150                     JSR     PC,TSTPRP           ;PREPARE DEVICE FOR TEST
     035570  054130                             .WORD   054130              ;TASK DESCRIPTOR AS FOLLOWS:
                                                                            ;CLEAR CONTROLLER & SELECT DEVICE
                                                                            ;VERIFY CONTROLLER CLEAR OPERATION
                                                                            ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                            ;VERIFY PACK ACKNOWLEDGE
                                                                            ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                            ;VERIFY RECALIBRATION
     035572  000404                             BR      80$                 ;GO TO 80$ IF NO ERROR
     035574  000240                             NOP                         ;RETURN HERE IF ERROR
     035576  104000                             EMT                         ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     035600  000137  036214                     JMP     200$                ;GO TO 200$ IF ERROR
1975 035604                             80$:
1976 035604  012703  000035                     MOV     #29.,R3             ;R3 = MAXIMUM SECTOR ADDRESS
1977 035610  012737  000000  001444             MOV     #0,RMDC0            ;CYLINDER = 0
1978 035616  012737  000000  001416             MOV     #0,RMDA0            ;TRACK = 0, SECTOR = 0
1979 035624  012737  000000  001442             MOV     #0,RMOF0            ;18 BIT FORMAT
1980 035632  012737  000005  001410  85$:       MOV     #SEEK!GO,RMCS10     ;SEEK COMMAND
1981 035640  012702  001543                     MOV     #PUTINX,R2          ;SETUP REGISTER INDEX TABLE
1982 035644  112722  000006                     MOVB    #RMDA,(R2)+         ;FOR SEEK TO CYLINDER 0
1983 035650  112722  000034                     MOVB    #RMDC,(R2)+
1984 035654  112722  000032                     MOVB    #RMOF,(R2)+
1985 035660  112722  000000                     MOVB    #RMCS1,(R2)+
1986 035664  112722  000200                     MOVB    #200,(R2)+
1987 035670  004737  044070                     JSR     PC,GETSTS           ;SETUP FOR STATUS FETCH
1988 035674  004737  044424                     JSR     PC,PUT              ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     035700  000404                             BR      90$                 ;GO TO 90$ IF NO ERROR
     035702  000240                             NOP                         ;RETURN HERE IF ERROR
     035704  104000                             EMT                         ;ERROR # DEFINED BY PUT SUBROUTINE
     035706  000137  036214                     JMP     200$                ;GO TO 200$ IF ERROR
1989 035712  004737  044766             90$:    JSR     PC,TIMOUT
1990
1991 035716  004737  044154                     JSR     PC,GET              ;GO READ REGISTER(S) WITH GET SUBROUTINE
     035722  000404                             BR      100$                ;GO TO 100$ IF NO ERROR
     035724  000240                             NOP                         ;RETURN HERE IF ERROR
     035726  104000                             EMT                         ;ERROR # DEFINED BY GET SUBROUTINE
     035730  000137  036214                     JMP     200$                ;GO TO 200$ IF ERROR
1992 035734                             100$:
1993 035734  004737  045152                     JSR     PC,PRIERR           ;GO CHECK FOR PRIMARY ERRORS
     035740  000405                             BR      110$                ;GO TO 110$ IF NO ERROR
     035742  000240                             NOP                         ;RETURN HERE IF ERROR
     035744  104000                             EMT                         ;ERROR # DEFINED BY PRIERR SUBROUTINE
     035746  004736                             JSR     PC,@(SP)+           ;GO BACK FOR MORE ERROR CHECKS
     035750  000137  036214                     JMP     200$                ;GO TO 200$ IF ERROR
1994 035754                             110$:
```

```
 1995 035754  004737  052270                 JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
      035760  000405                         BR      120$            ;GO TO 120$ IF NO ERROR
      035762  000240                         NOP                     ;RETURN HERE IF ERROR
      035764  104000                         EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
      035766  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      035770  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 1996 035774  012737  000031  001410  120$: MOV     #SEARCH!GO,RMCS1O       ;STORE SEARCH COMMAND
 1997 036002  004737  044424                 JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      036006  000404                         BR      130$            ;GO TO 130$ IF NO ERROR
      036010  000240                         NOP                     ;RETURN HERE IF ERROR
      036012  104000                         EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      036014  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 1998 036020                          130$:
 1999 036020  004737  044766                 JSR     PC,TIMOUT       ;WAIT FOR SEARCH TO COMPLETE
 2000
 2001 036024  004737  044154                 JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      036030  000404                         BR      140$            ;GO TO 140$ IF NO ERROR
      036032  000240                         NOP                     ;RETURN HERE IF ERROR
      036034  104000                         EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      036036  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 2002 036042                          140$:
 2003 036042  004737  045152                 JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      036046  000405                         BR      150$            ;GO TO 150$ IF NO ERROR
      036050  000240                         NOP                     ;RETURN HERE IF ERROR
      036052  104000                         EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      036054  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      036056  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 2004 036062                          150$:
 2005 036062  004737  057666                 JSR     PC,SCHSTS       ;GO VERIFY SEARCH OPERATION
      036066  000405                         BR      160$            ;GO TO 160$ IF NO ERROR
      036070  000240                         NOP                     ;RETURN HERE IF ERROR
      036072  104000                         EMT                     ;ERROR # DEFINED BY SCHSTS SUBROUTINE
      036074  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      036076  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 2006 036102                          160$:
 2007 036102  004737  051644                 JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
      036106  115760                         .WORD   NDTMSK          ;MASK FOR RMER1
      036110  000010                         .WORD   DPE             ;MASK FOR RMER2
      036112  000405                         BR      170$            ;GO TO 170$ IF NO ERROR
      036114  000240                         NOP                     ;RETURN HERE IF ERROR
      036116  104000                         EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
      036120  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      036122  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 2008 036126                          170$:
 2009 036126  004737  046004                 JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      036132  000405                         BR      180$            ;GO TO 180$ IF NO ERROR
      036134  000240                         NOP                     ;RETURN HERE IF ERROR
      036136  104000                         EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
      036140  004736                         JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      036142  000137  036214                 JMP     200$            ;GO TO 200$ IF ERROR
 2010 036146                          180$:
 2011 036146  005237  001416                 INC     RMDAO           ;INCREMENT SECTOR ADDRESS
 2012 036152  020337  001416                 CMP     R3,RMDAO        ;DONE ALL SECTORS??
 2013 036156  103306                         BHIS    120$            ;NO!!
 2014
 2015 036160  032737  010000  001442         BIT     #FMT16,RMOFO    ;IS 16 BIT FORMAT DONE??
 2016 036166  012737  010000  001442         MOV     #FMT16,RMOFO    ;SET 16 BIT FORMAT
```

```
2017 036174  001007                        BNE     200$            ;BR IF YES
2018 036176  012703  000037                MOV     #31.,R3         ;R3 = MAXIMUM SECTOR ADDRESS
2019 036202  012737  000000  001416        MOV     #0,RMDA0        ;START AT TRACK = 0, SECTOR = 0
2020 036210  000137  035632                JMP     85$             ;GO TEST AGAIN
2021 036214                          200$:
2022
2023                                 ;;*****************************************************************
                                     ;*TEST 62       SEARCH OFF CYLINDER
                                     ;;*****************************************************************
     036214                          TST62:
     036214  000004                        SCOPE                   ;SCOPE CALL
     036216  000240                        NOP                     ;START OF TEST
     036220  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     036224  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     036230  013701  001464                MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     036234  012737  000062  001226        MOV     #62,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2024
2025 036242  004737  043150                JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     036246  054130                        .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                  ;CLEAR CONTROLLER & SELECT DEVICE
                                                                  ;VERIFY CONTROLLER CLEAR OPERATION
                                                                  ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                  ;VERIFY PACK ACKNOWLEDGE
                                                                  ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                  ;VERIFY RECALIBRATION
     036250  000404                        BR      80$             ;GO TO 80$ IF NO ERROR
     036252  000240                        NOP                     ;RETURN HERE IF ERROR
     036254  104000                        EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     036256  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2026 036262                          80$:
2027 036262  012737  000000  001416        MOV     #0,RMDA0
2028 036270  012737  000000  001442        MOV     #0,RMOF0        ;START WITH 18 BIT MODE
2029 036276  012704  000035                MOV     #29.,R4         ;R4 = MAXIMUM SECTOR ADDRESS
2030 036302  012703  000632                MOV     #410.,R3        ;R3 = SEARCH CYLINDER
2031 036306  012705  000633                MOV     #411.,R5        ;R5 = SEEK CYLINDER
2032 036312  012737  000005  001410  90$:  MOV     #SEEK!GO,RMCS10 ;LOAD SEEK COMMAND
2033 036320  010537  001444                MOV     R5,RMDC0        ;LOAD CYLINDER ADDRESS
2034 036324  012702  001543                MOV     #PUTINX,R2      ;R2 POINTS TO REGISTER TABLE
2035 036330  112722  000032                MOVB    #RMOF,(R2)+     ;SETUP REGISTER INDEX TABLE
2036 036334  112722  000034                MOVB    #RMDC,(R2)+     ;FOR SEEK COMMAND
2037 036340  112722  000006                MOVB    #RMDA,(R2)+
2038 036344  112722  000000                MOVB    #RMCS1,(R2)+
2039 036350  112722  000200                MOVB    #200,(R2)+      ;TERMINATE TABLE
2040 036354  004737  044070                JSR     PC,GETSTS       ;SETUP FOR STATUS
2041 036360  004737  044424                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     036364  000404                        BR      100$            ;GO TO 100$ IF NO ERROR
     036366  000240                        NOP                     ;RETURN HERE IF ERROR
     036370  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     036372  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2042 036376                          100$:
2043 036376  004737  044766                JSR     PC,TIMOUT       ;WAIT FOR SEEK TO COMPLETE
2044
2045 036402  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     036406  000404                        BR      110$            ;GO TO 110$ IF NO ERROR
     036410  000240                        NOP                     ;RETURN HERE IF ERROR
     036412  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     036414  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
```

```
2046 036420                          110$:
2047 036420  004737  045152                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     036424  000405                        BR      120$            ;GO TO 120$ IF NO ERROR
     036426  000240                        NOP                     ;RETURN HERE IF ERROR
     036430  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     036432  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     036434  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2048 036440                          120$:
2049 036440  004737  052270                JSR     PC,SEKSTS       ;GO VERIFY RESULTS OF SEEK OPERATION
     036444  000405                        BR      130$            ;GO TO 130$ IF NO ERROR
     036446  000240                        NOP                     ;RETURN HERE IF ERROR
     036450  104000                        EMT                     ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     036452  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     036454  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2050 036460                          130$:
2051 036460  010337  001444                MOV     R3,RMDCO        ;LOAD CYLINDER ADDRESS
2052 036464  012737  000031  001410        MOV     #SEARCH.GO,RMCS1O       ;LOAD SEARCH COMMAND
2053 036472  004737  044424                JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     036476  000404                        BR      140$            ;GO TO 140$ IF NO ERROR
     036500  000240                        NOP                     ;RETURN HERE IF ERROR
     036502  104000                        EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     036504  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2054 036510                          140$:
2055 036510  004737  044766                JSR     PC,TIMOUT       ;WAIT FOR SEARCH TO COMPLETE
2056
2057 036514  004737  044154                JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     036520  000404                        BR      150$            ;GO TO 150$ IF NO ERROR
     036522  000240                        NOP                     ;RETURN HERE IF ERROR
     036524  104000                        EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     036526  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2058 036532                          150$:
2059 036532  004737  045152                JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     036536  000405                        BR      160$            ;GO TO 160$ IF NO ERROR
     036540  000240                        NOP                     ;RETURN HERE IF ERROR
     036542  104000                        EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     036544  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     036546  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2060 036552                          160$:
2061 036552  004737  057666                JSR     PC,SCHSTS       ;GO VERIFY SEARCH OPERATION
     036556  000405                        BR      170$            ;GO TO 170$ IF NO ERROR
     036560  000240                        NOP                     ;RETURN HERE IF ERROR
     036562  104000                        EMT                     ;ERROR # DEFINED BY SCHSTS SUBROUTINE
     036564  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     036566  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2062 036572                          170$:
2063 036572  004737  051644                JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     036576  115760                        .WORD   NDTMSK          ;MASK FOR RMER1
     036600  000010                        .WORD   DPE             ;MASK FOR RMER2
     036602  000405                        BR      180$            ;GO TO 180$ IF NO ERROR
     036604  000240                        NOP                     ;RETURN HERE IF ERROR
     036606  104000                        EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     036610  004736                        JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     036612  000137  036720                JMP     230$            ;GO TO 230$ IF ERROR
2064 036616                          180$:
2065 036616  004737  046004                JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     036622  000405                        BR      190$            ;GO TO 190$ IF NO ERROR
     036624  000240                        NOP                     ;RETURN HERE IF ERROR
```

```
        036626  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        036630  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        036632  000137  036720                  JMP     230$            ;GO TO 230$ IF ERROR
 2066   036636                          190$:
 2067   036636  005303                          DEC     R3              ;DECREMENT SEARCH CYLINDER
 2068   036640  005205                          INC     R5              ;INCREMENT SEEK CYLINDER
 2069   036642  005237  001416                  INC     RMDAO           ;INCREMENT SECTOR ADDRESS
 2070   036646  020437  001416                  CMP     R4,RMDAO        ;DONE ALL SECTORS??
 2071   036652  103020                          BHIS    225$            ;NO - DO NEXT SECTOR
 2072
 2073   036654  032737  010000  001442          BIT     #FMT16,RMOFO    ;DONE 16 BIT MODE YET ?
 2074   036662  001016                          BNE     230$            ;BR IF YES
 2075   036664  012737  010000  001442          MOV     #FMT16,RMOFO    ;SET 16 BIT FORMAT
 2076   036672  012704  000037                  MOV     #31.,R4         ;R4 = MAXIMUM SECTOR ADDRESS
 2077   036676  012703  000632                  MOV     #410.,R3        ;R3 = SEARCH CYLINDER
 2078   036702  012705  000633                  MOV     #411.,R5        ;R5 = SEEK CYLINDER
 2079   036706  012737  000000  001416          MOV     #0,RMDAO        ;START AT TRACK = 0, SECTOR - 0
 2080   036714  000137  036312          225$:   JMP     90$             ;GO TEST AGAIN
 2081   036720                          230$:
 2082
 2083                                   ;;*****************************************************************
                                        ;*TEST 63       SEARCH INVALID SECTOR
                                        ;;*****************************************************************
                                        ;;
        036720                          TST63:
        036720  000004                          SCOPE                   ;SCOPE CALL
        036722  000240                          NOP                     ;START OF TEST
        036724  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        036730  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        036734  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        036740  012737  000063  001226          MOV     #63,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2084
 2085   036746  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        036752  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        036754  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        036756  000240                          NOP                     ;RETURN HERE IF ERROR
        036760  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        036762  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
 2086   036766                          80$:
 2087   036766  012737  000000  001442          MOV     #0,RMOFO        ;SET 18 BIT FORMAT
 2088   036774  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
 2089   037002  012737  000036  001416          MOV     #30.,RMDAO      ;FIRST INVALID SECTOR = 30.
 2090   037010  012737  000031  001410          MOV     #SEARCH!GO,RMCS10       ;SEARCH COMMAND
 2091   037016  012702  001543                  MOV     #PUTINX,R2      ;WRITE REGISTER TABLE FOR
 2092   037022  112722  000034                  MOVB    #RMDC,(R2)+     ;COMMAND
 2093   037026  112722  000032                  MOVB    #RMOF,(R2)+
 2094   037032  112722  000006                  MOVB    #RMDA,(R2)+
 2095   037036  112722  000000                  MOVB    #RMCS1,(R2)+
 2096   037042  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
 2097   037046  004737  044070                  JSR     PC,GETSTS       ;SETUP STATUS FETCH
 2098   037052                          90$:
 2099   037052  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
```

```
        037056  000404                          BR      95$             ;GO TO 95$ IF NO ERROR
        037060  000240                          NOP                     ;RETURN HERE IF ERROR
        037062  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
        037064  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2100    037070                          95$:
2101    037070  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        037074  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
        037076  000240                          NOP                     ;RETURN HERE IF ERROR
        037100  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        037102  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2102    037106                          100$:
2103    037106  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
2104
2105    037112  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        037116  000404                          BR      120$            ;GO TO 120$ IF NO ERROR
        037120  000240                          NOP                     ;RETURN HERE IF ERROR
        037122  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        037124  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2106    037130                          120$:
2107    037130  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        037134  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
        037136  000240                          NOP                     ;RETURN HERE IF ERROR
        037140  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        037142  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037144  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2108    037150                          130$:
2109    037150  004737  057666                  JSR     PC,SCHSTS       ;GO VERIFY SEARCH OPERATION
        037154  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        037156  000240                          NOP                     ;RETURN HERE IF ERROR
        037160  104000                          EMT                     ;ERROR # DEFINED BY SCHSTS SUBROUTINE
        037162  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037164  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2110    037170                          140$:
2111    037170  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        037174  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        037176  000240                          NOP                     ;RETURN HERE IF ERROR
        037200  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        037202  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037204  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2112    037210                          150$:
2113    037210  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        037214  117760                          .WORD   IAE!NDTMSK              ;MASK FOR RMER1
        037216  000010                          .WORD   DPE             ;MASK FOR RMER2
        037220  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        037222  000240                          NOP                     ;RETURN HERE IF ERROR
        037224  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        037226  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037230  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2114    037234                          160$:
2115    037234  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        037240  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
        037242  000240                          NOP                     ;RETURN HERE IF ERROR
        037244  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        037246  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037250  000137  037320                  JMP     190$            ;GO TO 190$ IF ERROR
2116    037254                          170$:
2117    037254  005237  001416                  INC     RMDAD           ;INCREMENT SECTOR ADDRESS
```

```
2118 037260  022737  000037  001416          CMP     #31.,RMDAO      ;DONE ?
2119 037266  103012                          BHIS    175$            ;BR IF NO
2120
2121 037270  032737  010000  001442          BIT     #FMT16,RMOFO    ;TEST FOR 16 BIT MODE YET ?
2122 037276  001010                          BNE     190$            ;BR IF YES
2123 037300  012737  010000  001442          MOV     #FMT16,RMOFO    ;SET 16 BIT MODE
2124 037306  012737  000037  001416          MOV     #31.,RMDAO      ;SET INVALID SECTOR = 31.
2125 037314  000137  037052          175$:   JMP     90$             ;TEST NEXT SECTOR
2126 037320                          190$:
2127
2128                                 ;;*****************************************************************
                                     ;*TEST 64       SEARCH INVALID TRACK
                                     ;;*****************************************************************
     037320                          TST64:
     037320  000004                          SCOPE                   ;SCOPE CALL
     037322  000240                          NOP                     ;START OF TEST
     037324  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     037330  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     037334  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     037340  012737  000064  001226          MOV     #64,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2129
2130 037346  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     037352  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                             ;CLEAR CONTROLLER & SELECT DEVICE
                                             ;VERIFY CONTROLLER CLEAR OPERATION
                                             ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                             ;VERIFY PACK ACKNOWLEDGE
                                             ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                             ;VERIFY RECALIBRATION
     037354  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
     037356  000240                          NOP                     ;RETURN HERE IF ERROR
     037360  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     037362  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2131 037366                          80$:
2132 037366  012737  000031  001410          MOV     #SEARCH!GO,RMCS10       ;SEARCH COMMAND
2133 037374  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
2134 037402  013737  001332  001416          MOV     LSTRK,RMDAO     ;LAST TRACK, SECTOR - 0
2135 037410  105237  001417                  INCB    RMDAO+1         ;SETUP FIRST INVALID TRACK
2136 037414  012737  000000  001442          MOV     #0,RMOFO        ;SET 18 BIT FORMAT
2137 037422  012702  001543                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
2138 037426  112722  000006                  MOVB    #RMDA,(R2)+
2139 037432  112722  000034                  MOVB    #RMDC,(R2)+
2140 037436  112722  000032                  MOVB    #RMOF,(R2)+
2141 037442  112722  000000                  MOVB    #RMCS1,(R2)+
2142 037446  112722  000200                  MOVB    #200,(R2)+      ;TERMINATE TABLE
2143 037452  004737  044070                  JSR     PC,GETSTS
2144 037456                          90$:
2145 037456  004737  053530                  JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
     037462  000404                          BR      95$             ;GO TO 95$ IF NO ERROR
     037464  000240                          NOP                     ;RETURN HERE IF ERROR
     037466  104000                          EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
     037470  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2146 037474                          95$:
2147 037474  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     037500  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
     037502  000240                          NOP                     ;RETURN HERE IF ERROR
     037504  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
```

```
        037506  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2148    037512                          100$:
2149    037512  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
2150
2151    037516  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        037522  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
        037524  000240                          NOP                     ;RETURN HERE IF ERROR
        037526  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        037530  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2152    037534                          110$:
2153    037534  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        037540  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        037542  000240                          NOP                     ;RETURN HERE IF ERROR
        037544  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
        037546  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037550  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2154    037554                          120$:
2155    037554  004737  057666                  JSR     PC,SCHSTS       ;GO VERIFY SEARCH OPERATION
        037560  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
        037562  000240                          NOP                     ;RETURN HERE IF ERROR
        037564  104000                          EMT                     ;ERROR # DEFINED BY SCHSTS SUBROUTINE
        037566  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037570  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2156    037574                          130$:
2157    037574  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
        037600  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
        037602  000240                          NOP                     ;RETURN HERE IF ERROR
        037604  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
        037606  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037610  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2158    037614                          140$:
2159    037614  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        037620  117760                          .WORD   NDTMSK.IAE              ;MASK FOR RMER1
        037622  000010                          .WORD   DPE             ;MASK FOR RMER2
        037624  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        037626  000240                          NOP                     ;RETURN HERE IF ERROR
        037630  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        037632  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037634  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2160    037640                          150$:
2161    037640  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        037644  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
        037646  000240                          NOP                     ;RETURN HERE IF ERROR
        037650  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        037652  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        037654  000137  037730                  JMP     180$            ;GO TO 180$ IF ERROR
2162    037660                          160$:
2163    037660  105237  001417                  INCB    RMDAO+1         ;INCREMENT TRACK ADDRESS
2164    037664  123727  001417  000200          CMPB    RMDAO+1,#128.   ;DONE ALL TRACKS ?
2165    037672  101414                          BLOS    170$            ;NO!!
2166
2167    037674  032737  010000  001442          BIT     #FMT16,RMOFO    ;DONE 16 BIT MODE ?
2168    037702  001012                          BNE     180$            ;YES!!
2169    037704  012737  010000  001442          MOV     #FMT16,RMOFO    ;SET 16 BIT FORMAT
2170    037712  013737  001332  001416          MOV     LSTRK,RMDAO     ;LAST TRACK, SECTOR = 0
2171    037720  105237  001417                  INCB    RMDAO+1         ;SETUP FIRST INVALID TRACK ADDRESS
2172    037724  000137  037456          170$:   JMP     90$             ;TEST NEXT TRACK
```

```
2173
2174 037730                                  180$:
2175
2176                                         ;;*****************************************************************
                                             ;*TEST 65       SEARCH INVALID CYLINDER
                                             ;;*****************************************************************
     037730                                  TST65:
     037730  000004                                 SCOPE                   ;SCOPE CALL
     037732  000240                                 NOP                     ;START OF TEST
     037734  012706  001100                         MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     037740  013700  001276                         MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     037744  013701  001464                         MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     037750  012737  000065  00'226                 MOV     #65,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2177
2178 037756  004737  043150                         JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     037762  054130                               .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                         ;CLEAR CONTROLLER & SELECT DEVICE
                                                                         ;VERIFY CONTROLLER CLEAR OPERATION
                                                                         ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                         ;VERIFY PACK ACKNOWLEDGE
                                                                         ;RECALIBRATE IF ''SKI'' OR ''PIP'' IS SET
                                                                         ;VERIFY RECALIBRATION
     037764  000404                                 BR      80$             ;GO TO 80$ IF NO ERROR
     037766  000240                                 NOP                     ;RETURN HERE IF ERROR
     037770  104000                                 EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     037772  000137  040330                         JMP     190$            ;GO TO 190$ IF ERROR
2179 037776                                  80$:
2180 037776  012737  000031  001410                 MOV     #SEARCH!GO,RMCS1O       ;SEARCH COMMAND
2181 040004  012737  001467  001444                 MOV     #823.,RMDCO     ;START AT FIRST INVALID CYLINDER
2182 040012  012737  000000  001416                 MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
2183 040020  012737  000000  001442                 MOV     #0,RMOFO        ;SET 18 BIT FORMAT
2184 040026  012702  001543                         MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
2185 040032  112722  000032                         MOVB    #RMOF,(R2)+
2186 040036  112722  000034                         MOVB    #RMDC,(R2)+
2187 040042  112722  000006                         MOVB    #RMDA,(R2)+
2188 040046  112722  000000                         MOVB    #RMCS1,(R2)+
2189 040052  112722  000200                         MOVB    #200,(R2)+
2190 040056  004737  044070                         JSR     PC,GETSTS       ;SETUP STATUS FETCH
2191 040062                                  90$:
2192 040062  004737  053530                         JSR     PC,CNTCLR       ;GO ISSUE CONTROLLER CLEAR
     040066  000404                                 BR      95$             ;GO TO 95$ IF NO ERROR
     040070  000240                                 NOP                     ;RETURN HERE IF ERROR
     040072  104000                                 EMT                     ;ERROR NUMBER DEFINED BY SUBROUTINE
     040074  000137  040330                         JMP     190$            ;GO TO 190$ IF ERROR
2193 040100                                  95$:
2194 040100  004737  044424                         JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     040104  000404                                 BR      100$            ;GO TO 100$ IF NO ERROR
     040106  000240                                 NOP                     ;RETURN HERE IF ERROR
     040110  104000                                 EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
     040112  000137  040330                         JMP     190$            ;GO TO 190$ IF ERROR
2195 040116                                  100$:
2196 040116  004737  044766                         JSR     PC,TIMOUT
2197
2198 040122  004737  044154                         JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     040126  000404                                 BR      110$            ;GO TO 110$ IF NO ERROR
     040130  000240                                 NOP                     ;RETURN HERE IF ERROR
     040132  104000                                 EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
```

```
                040134  000137  040330                  JMP     190$            ;GO TO 190$ IF ERROR
2199            040140                          110$:
2200            040140  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
                040144  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
                040146  000240                          NOP                     ;RETURN HERE IF ERROR
                040150  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
                040152  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                040154  000137  040330                  JMP     190$            ;GO TO 190$ IF ERROR
2201            040160                          120$:
2202            040160  004737  057666                  JSR     PC,SCHSTS       ;GO VERIFY SEARCH OPERATION
                040164  000405                          BR      130$            ;GO TO 130$ IF NO ERROR
                040166  000240                          NOP                     ;RETURN HERE IF ERROR
                040170  104000                          EMT                     ;ERROR # DEFINED BY SCHSTS SUBROUTINE
                040172  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                040174  000137  040330                  JMP     190$            ;GO TO 190$ IF ERROR
2203            040200                          130$:
2204            040200  004737  061232                  JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
                040204  000405                          BR      140$            ;GO TO 140$ IF NO ERROR
                040206  000240                          NOP                     ;RETURN HERE IF ERROR
                040210  104000                          EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
                040212  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                040214  000137  040330                  JMP     190$            ;GO TO 190$ IF ERROR
2205            040220                          140$:
2206            040220  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
                040224  117760                          .WORD   IAE!NDTMSK              ;MASK FOR RMER1
                040226  000010                          .WORD   DPE             ;MASK FOR RMER2
                040230  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
                040232  000240                          NOP                     ;RETURN HERE IF ERROR
                040234  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
                040236  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                040240  000137  040330                  JMP     190$            ;GO TO 190$ IF ERROR
2207            040244                          150$:
2208            040244  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
                040250  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
                040252  000240                          NOP                     ;RETURN HERE IF ERROR
                040254  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                040256  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
                040260  000137  040330                  JMP     190$            ;GO TO 190$ IF ERROR
2209            040264                          160$:
2210            040264  005237  001444                  INC     RMDCO           ;INCREMENT CYLINDER ADDRESS
2211            040270  023727  001444  002000          CMP     RMDCO,#1024.    ;DONE ALL CYLINDERS ?
2212            040276  002412                          BLT     170$            ;NO!!
2213
2214            040300  032737  010000  001442          BIT     #FMT16,RMOFO    ;SET 16 BIT FORMAT
2215            040306  001010                          BNE     190$            ;YES!!
2216            040310  012737  010000  001442          MOV     #FMT16,RMOFO    ;SET 16 BIT FORMAT
2217            040316  012737  001467  001444          MOV     #823.,RMDCO     ;START AT FIRST INVALID CYLINDER
2218            040324  000137  040062          170$:   JMP     90$
2219
2220            040330                          190$:
2221
2222                                            ;;*****************************************************************
                                                ;*TEST 66        IVC SEARCH TEST
                                                ;;*****************************************************************
                040330                          TST66:
                040330  000004                          SCOPE                   ;SCOPE CALL
                040332  000240                          NOP                     ;START OF TEST
```

```
        040334  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        040340  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        040344  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        040350  012737  000066  001226          MOV     #66,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  2223
  2224  040356  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        040362  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        040364  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        040366  000240                          NOP                     ;RETURN HERE IF ERROR
        040370  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        040372  000137  040710                  JMP     160$            ;GO TO 160$ IF ERROR
  2225  040376                          80$:
  2226  040376  112737  000024  001543          MOVB    #RMMR1,PUTINX   ;SETUP PUT INDEX TABLE
        040404  112737  000200  001544          MOVB    #200,PUTINX+1   ;SET TERMINATOR BYTE
        040412  012737  000001  001434          MOV     #DMD,RMMR10     ;SET RMMR1 OUTPUT BUFFER = DMD
        040420  004737  044424                  JSR     PC,PUT          ;GO WRITE RMMR1 VIA PUT SUBROUTINE
        040424  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
        040426  000240                          NOP                     ;RETURN HERE IF ERROR
        040430  104000                          EMT                     ;ERROR DEFINED BY PUT SUBROUTINE
        040432  000137  040710                  JMP     160$            ;GO TO 160$ IF ERROR
  2227  040436  012737  000000  001434  90$:    MOV     #0,RMMR10       ;RESET DIAGNOSTIC MODE
  2228  040444  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER = 0
  2229  040452  012737  000000  001416          MOV     #0,RMDA0        ;TRACK = 0, SECTOR = 0
  2230  040460  012737  000000  001442          MOV     #0,RMOF0        ;18 BIT FORMAT
  2231  040466  012737  000031  001410          MOV     #SEARCH!GO,RMCS10       ;SEARCH COMMAND
  2232  040474  012702  001543                  MOV     #PUTINX,R2      ;LOAD INDEX TABLE
  2233  040500  112722  000024                  MOVB    #RMMR1,(R2)+
  2234  040504  112722  000034                  MOVB    #RMDC,(R2)+
  2235  040510  112722  000006                  MOVB    #RMDA,(R2)+
  2236  040514  112722  000032                  MOVB    #RMOF,(R2)+
  2237  040520  112722  000000                  MOVB    #RMCS1,(R2)+
  2238  040524  112722  000200                  MOVB    #200,(R2)+
  2239  040530  004737  044070                  JSR     PC,GETSTS
  2240  040534  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
        040540  000404                          BR      100$            ;GO TO 100$ IF NO ERROR
        040542  000240                          NOP                     ;RETURN HERE IF ERROR
        040544  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
        040546  000137  040710                  JMP     160$            ;GO TO 160$ IF ERROR
  2241  040552                          100$:
  2242  040552  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
  2243
  2244  040556  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
        040562  000404                          BR      110$            ;GO TO 110$ IF NO ERROR
        040564  000240                          NOP                     ;RETURN HERE IF ERROR
        040566  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
        040570  000137  040710                  JMP     160$            ;GO TO 160$ IF ERROR
  2245  040574                          110$:
  2246  040574  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
        040600  000405                          BR      120$            ;GO TO 120$ IF NO ERROR
        040602  000240                          NOP                     ;RETURN HERE IF ERROR
        040604  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
```

```
        040606  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        040610  000137  040710                  JMP     160$            ;GO TO 160$ IF ERROR
2247    040614                          120$:
2248    040614  032737  010000  001376          BIT     #IVC,RMER21     ;DID IVC SET??
2249    040622  001012                          BNE     130$            ;YES!!
2250    040624  012737  010000  001140          MOV     #IVC,$GDDAT     ;EXPECTED STATUS
2251    040632  013737  001376  001142          MOV     RMER21,$BDDAT   ;RECEIVED STATUS
2252    040640  042737  167777  001142          BIC     #^CIVC,$BDDAT
2253    040646  104260                          EMT     260
2254    040650                          130$:
2255    040650  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
        040654  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
        040656  010010                          .WORD   IVC.DPE         ;MASK FOR RMER2
        040660  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
        040662  000240                          NOP                     ;RETURN HERE IF ERROR
        040664  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
        040666  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
        040670  000137  040710                  JMP     160$            ;GO TO 160$ IF ERROR
2256    040674                          150$:
2257    040674  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
        040700  000403                          BR      160$            ;GO TO 160$ IF NO ERROR
        040702  000240                          NOP                     ;RETURN HERE IF ERROR
        040704  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
        040706  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
2258    040710                          160$:
2259
2260                                    ;;*****************************************************************
                                        ;*TEST 67        ABORT SEARCH TEST
                                        ;;*****************************************************************
        040710                          TST67:
        040710  000004                          SCOPE                   ;SCOPE CALL
        040712  000240                          NOP                     ;START OF TEST
        040714  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
        040720  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
        040724  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
        040730  012737  000067  001226          MOV     #67,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2261
2262    040736  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
        040742  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                        ;CLEAR CONTROLLER & SELECT DEVICE
                                                                        ;VERIFY CONTROLLER CLEAR OPERATION
                                                                        ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                        ;VERIFY PACK ACKNOWLEDGE
                                                                        ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                        ;VERIFY RECALIBRATION
        040744  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
        040746  000240                          NOP                     ;RETURN HERE IF ERROR
        040750  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
        040752  000137  041254                  JMP     160$            ;GO TO 160$ IF ERROR
2263    040756                          80$:
2264    040756  012737  040000  001424          MOV     #UNS,RMER10     ;SET UNSAFE ERROR
2265    040764  012737  000031  001410          MOV     #SEARCH!GO,RMCS10       ;SEARCH COMMAND
2266    040772  012737  000000  001444          MOV     #0,RMDC0        ;CYLINDER 0
2267    041000  012737  000000  001416          MOV     #0,RMDA0        ;SECTOR = 0, TRACK = 0
2268    041006  012702  001543                  MOV     #PUTINX,R2      ;WRITE REGISTER INDEX TABLE
2269    041012  112722  000034                  MOVB    #RMDC,(R2)+
2270    041016  112722  000006                  MOVB    #RMDA,(R2)+
```

```
 2271 041022  112722  000014                    MOVB    #RMER1,(R2)+
 2272 041026  112722  000000                    MOVB    #RMCS1,(R2)+
 2273 041032  112722  000200                    MOVB    #200,(R2)+      ;TERMINATE TABLE
 2274 041036  004737  044070                    JSR     PC,GETSTS       ;SETUP FOR STATUS
 2275 041042  004737  044424                    JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
      041046  000404                            BR      90$             ;GO TO 90$ IF NO ERROR
      041050  000240                            NOP                     ;RETURN HERE IF ERROR
      041052  104000                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
      041054  000137  041254                    JMP     160$            ;GO TO 160$ IF ERROR
 2276 041060                            90$:
 2277 041060  004737  044154                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      041064  000404                            BR      100$            ;GO TO 100$ IF NO ERROR
      041066  000240                            NOP                     ;RETURN HERE IF ERROR
      041070  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      041072  000137  041254                    JMP     160$            ;GO TO 160$ IF ERROR
 2278 041076                            100$:
 2279 041076  032737  020000  001346            BIT     #PIP,RMDSI      ;IS PIP SET??
 2280 041104  001412                            BEQ     110$            ;NO!!
 2281 041106  012737  000000  001140            MOV     #0,$GDDAT       ;EXPECTED STATUS
 2282 041114  013737  001346  001142            MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
 2283 041122  042737  157777  001142            BIC     #^CPIP,$BDDAT
 2284 041130  104261                            EMT     261
 2285 041132                            110$:
 2286 041132  004737  044766                    JSR     PC,TIMOUT       ;WAIT FOR GO TO RESET
 2287
 2288 041136  004737  044154                    JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
      041142  000404                            BR      120$            ;GO TO 120$ IF NO ERROR
      041144  000240                            NOP                     ;RETURN HERE IF ERROR
      041146  104000                            EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
      041150  000137  041254                    JMP     160$            ;GO TO 160$ IF ERROR
 2289 041154                            120$:
 2290 041154  004737  045152                    JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
      041160  000405                            BR      130$            ;GO TO 130$ IF NO ERROR
      041162  000240                            NOP                     ;RETURN HERE IF ERROR
      041164  104000                            EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
      041166  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      041170  000137  041254                    JMP     160$            ;GO TO 160$ IF ERROR
 2291 041174                            130$:
 2292 041174  004737  061232                    JSR     PC,STCDRVSTS    ;GO CHECK FOR CHANGES IN DRIVE STATUS
      041200  000405                            BR      140$            ;GO TO 140$ IF NO ERROR
      041202  000240                            NOP                     ;RETURN HERE IF ERROR
      041204  104000                            EMT                     ;ERROR # DEFINED BY STCDRVSTS SUBROUTINE
      041206  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      041210  000137  041254                    JMP     160$            ;GO TO 160$ IF ERROR
 2293 041214                            140$:
 2294 041214  004737  051644                    JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
      041220  155760                            .WORD   NDTMSK!UNS              ;MASK FOR RMER1
      041222  000010                            .WORD   DPE             ;MASK FOR RMER2
      041224  000405                            BR      150$            ;GO TO 150$ IF NO ERROR
      041226  000240                            NOP                     ;RETURN HERE IF ERROR
      041230  104000                            EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
      041232  004736                            JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
      041234  000137  041254                    JMP     160$            ;GO TO 160$ IF ERROR
 2295 041240                            150$:
 2296 041240  004737  046004                    JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
      041244  000403                            BR      160$            ;GO TO 160$ IF NO ERROR
      041246  000240                            NOP                     ;RETURN HERE IF ERROR
```

```
                041250  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
                041252  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
2297            041254                          160$:
2298
2299                                            ;;*****************************************************************
                                                ;*TEST 70        SEARCH AT OFFSET
                                                ;;*****************************************************************
                041254                          TST70:
                041254  000004                          SCOPE                   ;SCOPE CALL
                041256  000240                          NOP                     ;START OF TEST
                041260  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
                041264  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
                041270  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
                041274  012737  000070  001226          MOV     #70,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2300
2301            041302  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
                041306  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
                                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                                ;VERIFY RECALIBRATION
                041310  000404                          BR      80$             ;GO TO 80$ IF NO ERROR
                041312  000240                          NOP                     ;RETURN HERE IF ERROR
                041314  104000                          EMT                     ;ERROR # DEFINED BY TSTPRP SUBROUTINE
                041316  000137  041612                  JMP     180$            ;GO TO 180$ IF ERROR
2302            041322                          80$:
2303            041322  012737  000000  001444          MOV     #0,RMDCO        ;CYLINDER = 0
2304            041330  012737  000000  001416          MOV     #0,RMDAO        ;TRACK = 0, SECTOR = 0
2305            041336  012737  010000  001442          MOV     #FMT16,RMOFO    ;16 BIT FORMAT
2306            041344  012737  000015  001410          MOV     #OFFSET!GO,RMCS10       ;OFFSET COMMAND
2307            041352  012702  001543                  MOV     #PUTINX,R2      ;SETUP REGISTER INDEX TABLE
2308            041356  112722  000006                  MOVB    #RMDA,(R2)+     ;FOR SEEK TO CYLINDER 0
2309            041362  112722  000034                  MOVB    #RMDC,(R2)+
2310            041366  112722  000032                  MOVB    #RMOF,(R2)+
2311            041372  112722  000000                  MOVB    #RMCS1,(R2)+
2312            041376  112722  000200                  MOVB    #200,(R2)+
2313            041402  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                041406  000404                          BR      90$             ;GO TO 90$ IF NO ERROR
                041410  000240                          NOP                     ;RETURN HERE IF ERROR
                041412  1040                            EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                041414  0001    041612                  JMP     180$            ;GO TO 180$ IF ERROR
2314            041420                          90$:
2315            041420  004737  044070                  JSR     PC,GETSTS       ;SETUP FOR STATUS FETCH
2316            041424  004737  044766                  JSR     PC,TIMOUT
2317            041430                          120$:
2318            041430  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;LOAD REGISTER INDEX TABLE
2319            041436  112737  000200  001544          MOVB    #200,PUTINX+1
2320            041444  012737  000031  001410          MOV     #SEARCH!GO,RMCS10       ;STORE SEARCH COMMAND
2321            041452  004737  044424                  JSR     PC,PUT          ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
                041456  000404                          BR      130$            ;GO TO 130$ IF NO ERROR
                041460  000240                          NOP                     ;RETURN HERE IF ERROR
                041462  104000                          EMT                     ;ERROR # DEFINED BY PUT SUBROUTINE
                041464  000137  041612                  JMP     180$            ;GO TO 180$ IF ERROR
2322            041470                          130$:
2323            041470  004737  044766                  JSR     PC,TIMOUT       ;WAIT FOR SEARCH TO COMPLETE
```

```
2324
2325 041474  004737  044154                  JSR     PC,GET          ;GO READ REGISTER(S) WITH GET SUBROUTINE
     041500  000404                          BR      140$            ;GO TO 140$ IF NO ERROR
     041502  000240                          NOP                     ;RETURN HERE IF ERROR
     041504  104000                          EMT                     ;ERROR # DEFINED BY GET SUBROUTINE
     041506  000137  041612                  JMP     180$            ;GO TO 180$ IF ERROR
2326 041512                          140$:
2327 041512  004737  045152                  JSR     PC,PRIERR       ;GO CHECK FOR PRIMARY ERRORS
     041516  000405                          BR      150$            ;GO TO 150$ IF NO ERROR
     041520  000240                          NOP                     ;RETURN HERE IF ERROR
     041522  104000                          EMT                     ;ERROR # DEFINED BY PRIERR SUBROUTINE
     041524  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     041526  000137  041612                  JMP     180$            ;GO TO 180$ IF ERROR
2328 041532                          150$:
2329 041532  004737  057666                  JSR     PC,SCHSTS       ;GO VERIFY SEARCH OPERATION
     041536  000405                          BR      160$            ;GO TO 160$ IF NO ERROR
     041540  000240                          NOP                     ;RETURN HERE IF ERROR
     041542  104000                          EMT                     ;ERROR # DEFINED BY SCHSTS SUBROUTINE
     041544  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     041546  000137  041612                  JMP     180$            ;GO TO 180$ IF ERROR
2330 041552                          160$:
2331 041552  004737  051644                  JSR     PC,CMPERRSTS    ;CHECK ANY ERRORS NOT MASKED
     041556  115760                          .WORD   NDTMSK          ;MASK FOR RMER1
     041560  000010                          .WORD   DPE             ;MASK FOR RMER2
     041562  000405                          BR      170$            ;GO TO 170$ IF NO ERROR
     041564  000240                          NOP                     ;RETURN HERE IF ERROR
     041566  104000                          EMT                     ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     041570  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
     041572  000137  041612                  JMP     180$            ;GO TO 180$ IF ERROR
2332 041576                          170$:
2333 041576  004737  046004                  JSR     PC,SECERR       ;GO CHECK FOR SECONDARY ERRORS
     041602  000403                          BR      180$            ;GO TO 180$ IF NO ERROR
     041604  000240                          NOP                     ;RETURN HERE IF ERROR
     041606  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
     041610  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
2334 041612                          180$:
2335
2343                                        ;;*****************************************************************
                                            ;*TEST 71        HEAD ALIGNMENT SEEK
                                            ;*
                                            ;*HEAD ALIGNMENT FOR AN RM05/3/2 DRIVE, IS PERFORMED AT CYLINLER
                                            ;*245 DECIMAL USING THE CE PACK. A 60 MINUTE WARMUP IS REQUIRED IF
                                            ;*THE PACK WAS NOT IN THE DRIVE PRIOR TO HEAD ALIGNMENT OPERATION.
                                            ;*
                                            ;;*****************************************************************
     041612                                 TST71:
     041612  000004                          SCOPE                   ;SCOPE CALL
     041614  000240                          NOP                     ;START OF TEST
     041616  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
     041622  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
     041626  013701  001464                  MOV     TSTQUE,R1       ;(R1) = DEVICE BEING TESTED
     041632  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
     041640  012737  000071  001226          MOV     #71,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2344
2345 041646                          10$:
2346 041646  004737  043150                  JSR     PC,TSTPRP       ;PREPARE DEVICE FOR TEST
     041652  054130                          .WORD   054130          ;TASK DESCRIPTOR AS FOLLOWS:
```

```
                                                                ;CLEAR CONTROLLER & SELECT DEVICE
                                                                ;VERIFY CONTROLLER CLEAR OPERATION
                                                                ;PACK ACKNOWLEDGE IF VOLUME NOT VALID
                                                                ;VERIFY PACK ACKNOWLEDGE
                                                                ;RECALIBRATE IF "SKI" OR "PIP" IS SET
                                                                ;VERIFY RECALIBRATION
     041654  000404                          BR      80$       ;GO TO 80$ IF NO ERROR
     041656  000240                          NOP               ;RETURN HERE IF ERROR
     041660  104000                          EMT               ;ERROR # DEFINED BY TSTPRP SUBROUTINE
     041662  000137  042100                  JMP     140$      ;GO TO 140$ IF ERROR
2347 041666                          80$:
2348 041666  012737  000365  001444          MOV     #245.,RMDC0       ;CYLINDER = 245.
2349 041674  012737  000000  001416          MOV     #0,RMDA0          ;TRACK = 0, SECTOR = 0
2350 041702  012737  000005  001410          MOV     #SEEK!GO,RMCS10   ;LOAD SEEK COMMAND IN BUFFER
2351 041710  012702  001543                  MOV     #PUTINX,R2        ;R2 POINTS TO REGISTER TABLE
2352 041714  112722  000034                  MOVB    #RMDC,(R2)+       ;WRITE REGISTER INDEX TABLE
2353 041720  112722  000006                  MOVB    #RMDA,(R2)+
2354 041724  112722  000000                  MOVB    #RMCS1,(R2)+
2355 041730  112722  000200                  MOVB    #200,(R2)+        ;WRITE TERMINATOR
2356 041734  004737  044424                  JSR     PC,PUT            ;GO WRITE REGISTER(S) WITH PUT SUBROUTINE
     041740  000404                          BR      90$               ;GO TO 90$ IF NO ERROR
     041742  000240                          NOP                       ;RETURN HERE IF ERROR
     041744  104000                          EMT                       ;ERROR # DEFINED BY PUT SUBROUTINE
     041746  000137  042100                  JMP     140$              ;GO TO 140$ IF ERROR
2357 041752  004737  044070          90$:    JSR     PC,GETSTS         ;SETUP FOR STATUS FETCH
2358 041756  004737  044766                  JSR     PC,TIMOUT         ;WAIT FOR SEEK TO COMPLETE
2359
2360 041762  004737  044154                  JSR     PC,GET            ;GO READ REGISTER(S) WITH GET SUBROUTINE
     041766  000404                          BR      100$              ;GO TO 100$ IF NO ERROR
     041770  000240                          NOP                       ;RETURN HERE IF ERROR
     041772  104000                          EMT                       ;ERROR # DEFINED BY GET SUBROUTINE
     041774  000137  042100                  JMP     140$              ;GO TO 140$ IF ERROR
2361 042000                          100$:
2362 042000  004737  045152                  JSR     PC,PRIERR         ;GO CHECK FOR PRIMARY ERRORS
     042004  000405                          BR      110$              ;GO TO 110$ IF NO ERROR
     042006  000240                          NOP                       ;RETURN HERE IF ERROR
     042010  104000                          EMT                       ;ERROR # DEFINED BY PRIERR SUBROUTINE
     042012  004736                          JSR     PC,@(SP)+         ;GO BACK FOR MORE ERROR CHECKS
     042014  000137  042100                  JMP     140$              ;GO TO 140$ IF ERROR
2363 042020                          110$:
2364 042020  004737  052270                  JSR     PC,SEKSTS         ;GO VERIFY RESULTS OF SEEK OPERATION
     042024  000405                          BR      120$              ;GO TO 120$ IF NO ERROR
     042026  000240                          NOP                       ;RETURN HERE IF ERROR
     042030  104000                          EMT                       ;ERROR # DEFINED BY SEKSTS SUBROUTINE
     042032  004736                          JSR     PC,@(SP)+         ;GO BACK FOR MORE ERROR CHECKS
     042034  000137  042100                  JMP     140$              ;GO TO 140$ IF ERROR
2365 042040                          120$:
2366 042040  004737  051644                  JSR     PC,CMPERRSTS      ;CHECK ANY ERRORS NOT MASKED
     042044  115760                          .WORD   NDTMSK            ;MASK FOR RMER1
     042046  000010                          .WORD   DPE               ;MASK FOR RMER2
     042050  000405                          BR      130$              ;GO TO 130$ IF NO ERROR
     042052  000240                          NOP                       ;RETURN HERE IF ERROR
     042054  104000                          EMT                       ;ERROR # DEFINED BY CMPERRSTS SUBROUTINE
     042056  004736                          JSR     PC,@(SP)+         ;GO BACK FOR MORE ERROR CHECKS
     042060  000137  042100                  JMP     140$              ;GO TO 140$ IF ERROR
2367 042064                          130$:
2368 042064  004737  046004                  JSR     PC,SECERR         ;GO CHECK FOR SECONDARY ERRORS
```

```
042070  000403                          BR      140$            ;GO TO 140$ IF NO ERROR
042072  000240                          NOP                     ;RETURN HERE IF ERROR
042074  104000                          EMT                     ;ERROR # DEFINED BY SECERR SUBROUTINE
042076  004736                          JSR     PC,@(SP)+       ;GO BACK FOR MORE ERROR CHECKS
2369 042100                     140$:
2370
2374
```

```
                                        .SBTTL  END OF SUB-PASS ROUTINE

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

042100  000004                  $EOSP:  SCOPE
042102  000240                          NOP
042104  013700  001464                  MOV     TSTQUE,R0       ;GET POINTER TO TSTQUE
042110  062700  000002                  ADD     #2,R0           ;ADJUST POINTER TO NEXT DEVICE
042114  010037  001464                  MOV     R0,TSTQUE       ;SAVE POINTER TO TSTQUE
042120  005710                          TST     (R0)            ;ANY MORE DEVICES FOR TEST ?
042122  001402                          BEQ     1$              ;BR IF NO
042124  000137  007756                  JMP     READY           ;YES, JUMP TO READY
042130  012737  001466  001464  1$:     MOV     #TSTQUE+2,TSTQUE        ;INITIALIZE POINTER TO FIRST DEVICE IN
                                                                ;TEST QUE TABLE

                                        .SBTTL  END OF PASS ROUTINE

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

042136                            $EOP:
042136  000240                          NOP
042140  005037  001116                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
042144  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
042150  005237  001230                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
042154  042737  100000  001230          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
042162  005327                          DEC     (PC)+           ;;LOOP?
042164  000001                $EOPCT:   .WORD   1
042166  003063                          BGT     $DOAGN          ;;YES
042170  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
042172  000001                $ENDCT:   .WORD   1
042174  042164                          $EOPCT
042176  104401  042204                  TYPE    ,65$            ;;TYPE ASCIZ STRING
042202  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
042222                          64$:
042222  013746  001230                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
042226  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
042230  104401  042236                  TYPE    ,67$            ;;TYPE ASCIZ STRING
042234  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
042300                          66$:
042300  013746  001126                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
042304  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
042306  104401  001217                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
042312  005037  001126                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
042316  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
```

```
042322  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
042324  000005                          RESET                   ;;CLEAR THE WORLD
042326  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
042330  000240                          NOP                     ;;SAVE ROOM
042332  000240                          NOP                     ;;FOR
042334  000240                          NOP                     ;;ACT11
042336                          $DOAGN:
042336  000137                          JMP     @(PC)+          ;;RETURN
042340  007756                  $RTNAD: .WORD   READY
042342     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
                                        .EVEN
```

21

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

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

042346                                  ERRTYP:
042346  104414                                  SAVREG
042350  032777  020000  136576                  BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
042356  001402                                  BEQ     1$              ;NO!!
042360  000137  043076                          JMP     21$             ;YES.!
                                        ;TYPE UNIT NUMBER, TEST NUMBER, ERROR NUMBER, AND PROGRAM COUNTER
042364  104401  001217                1$:       TYPE    ,$CRLF
042370  104401  043112                          TYPE    ,ERTY00         ;TYPE "UNT#"
042374  013746  001234                          MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                        ;;TYPE UNIT NUMBER
042400  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
042402     003                                  .BYTE   3               ;;TYPE 3 DIGIT(S)
042403     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
042404  005037  043102                          CLR     TSTNMB          ;LOAD TEST NUMBER FOR
042410  013737  001226  043102                  MOV     $TESTN,TSTNMB
042416  104401  043120                          TYPE    ,ERTY01         ;TYPE "TST#"
042422  013746  043102                          MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                        ;;TYPE TEST NUMBER
042426  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
042430     003                                  .BYTE   3               ;;TYPE 3 DIGIT(S)
042431     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
042432  005037  043104                          CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
042436  113737  001130  043104                  MOVB    $ITEMB,ERRNMB   ;TYPEOUT
042444  001406                                  BEQ     2$              ;SKIP IF NO ERROR CALLED
042446  104401  043130                          TYPE    ,ERTY02         ;TYPE "ERR#"
042452  013746  043104                          MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                        ;;TYPE ERROR NUMBER
042456  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
042460     003                                  .BYTE   3               ;;TYPE 3 DIGIT(S)
042461     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
042462  104401  043137                2$:       TYPE    ,ERTY03         ;TYPE "PC="
042466  013746  001132                          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;TYPE PROGRAM COUNTER
042472  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
042474     006                                  .BYTE   6               ;;TYPE 6 DIGIT(S)
042475     001                                  .BYTE   1               ;;TYPE LEADING ZEROS
                                        ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
042476  005737  043104                3$:       TST     ERRNMB          ;WAS AN ERROR CALLED?
042502  001575                                  BEQ     21$             ;NO!!
042504  104401  001217                          TYPE    ,$CRLF          ;YES-TYPE CRLF
042510  105037  043110                          CLRB    BOTFLG          ;CLEAR BOT FLAG
042514  105037  043111                          CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
042520  013700  043104                          MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
```

```
042524  006300                          ASL     R0              ;FOUR ENTRIES IN ERROR
042526  006300                          ASL     R0              ;TABLE
042530  006300                          ASL     R0
042532  062700  001562                  ADD     #$ERRTB-8.,R0
042536  011001                          MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
                                                                ;TABLE
042540  001507                          BEQ     13$             ;BRANCH IF NO ERROR MESSAGE
                                ;TYPE THE ERROR MESSAGE
042542  012102                  4$:     MOV     (R1)+,R2        ;R2 = ADDRESS OF MESSAGE STRING
042544  001505                          BEQ     12$             ;BRANCH IF END OF MESSAGE
042546  010237  042714                  MOV     R2,11$          ;LOAD ADDRESS OF STRING
042552  005037  043106                  CLR     BOTADR          ;CLEAR BOT ADDRESS
042556  112203                  5$:     MOVB    (R2)+,R3        ;END OF STRING??
042560  001454                          BEQ     10$             ;YES!!
042562  122703  000015                  CMPB    #CR,R3          ;CARRIAGE RETURN??
042566  001003                          BNE     6$              ;NO!!
042570  105037  043111                  CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
042574  000770                          BR      5$              ;GET NEXT CHARACTER
042576  122703  000012          6$:     CMPB    #LF,R3          ;LINE FEED??
042602  001765                          BEQ     5$              ;YES-GET NEXT CHARACTER
042604  122703  000011                  CMPB    #HT,R3          ;HORIZONTAL TAB??
042610  001007                          BNE     8$              ;NO!!
042612  105237  043111          7$:     INCB    CHRCNT          ;ADJUST CHARACTER COUNT
042616  132737  000007  043111          BITB    #7,CHRCNT
042624  001372                          BNE     7$
042626  000407                          BR      9$
042630  105237  043111          8$:     INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
042634  122703  000040                  CMPB    #' ,R3          ;SPACE??
042640  001002                          BNE     9$              ;NO!!
042642  010237  043106                  MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
042646  122737  000100  043111  9$:     CMPB    #64.,CHRCNT     ;END OF LINE??
042654  103340                          BHIS    5$              ;NO!!
042656  013704  043106                  MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
042662  001007                          BNE     90$             ;BRANCH IF SPACE DETECTED
042664  104401  001217                  TYPE    ,$CRLF          ;TYPE CRLF
042670  105037  043111                  CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
042674  013702  042714                  MOV     11$,R2          ;SET UP R2 FOR TESTING
042700  000726                          BR      5$
042702  105044                  90$:    CLRB    -(R4)           ;REPLACE SPACE
042704  112737  177777  043110          MOVB    #-1,BOTFLG      ;SET BOT FLAG
042712  104401                  10$:    TYPE                    ;TYPE ERROR MESSAGE STRING
042714  000000                  11$:    .WORD                   ;STRING ADDRESS GOES HERE
042716  105737  043110                  TSTB    BOTFLG          ;WAS STRING TRUNCATED??
042722  001707                          BEQ     4$              ;NO!!
042724  104401  001217                  TYPE    ,$CRLF          ;YES-TYPE CRLF
042730  105037  043110                  CLRB    BOTFLG          ;CLEAR BOT FLAG
042734  105037  043111                  CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
042740  013702  043106                  MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
042744  010237  042714                  MOV     R2,11$          ;SETUP 11$ FOR TYPING
042750  112742  000040                  MOVB    #' ,-(R2)       ;RESTORE SPACE
042754  105722                          TSTB    (R2)+           ;RESTORE R2
042756  000677                          BR      5$              ;TYPE REST OF STRING
042760                          12$:
                                ;TYPE ERROR HEADER AND ERROR DATA
042760                          13$:
042760  016001  000002                  MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
042764  001444                          BEQ     21$             ;BRANCH IF NO HEADER
```

```
 99 042766  104401  001217                TYPE    ,$CRLF          ;(ASSUME NO DATA)
100 042772  016002  000004                MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
101 042776  016003  000006                MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
102 043002  012137  043012        14$:    MOV     (R1)+,15$       ;PUT HEADER ADDRESS FOR TYPE
103 043006  001433                        BEQ     21$             ;BRANCH IF END OF HEADERS
104                                                               ;(ASSUME END OF DATA)
105 043010  104401                        TYPE
106 043012  000000                15$:    .WORD   0               ;HEADER ADDRESS GOES HERE
107 043014  104401  001217                TYPE    ,$CRLF
108 043020  005702                        TST     R2              ;DATA WITH HEADER??
109 043022  001767                        BEQ     14$             ;NO!!
110 043024  012204                        MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
111 043026  012305                        MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
112 043030  105725                16$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
113 043032  100407                        BMI     18$             ;BINARY
114 043034  001403                        BEQ     17$             ;OCTAL
115 043036  013446                        MOV     @(R4)+,-(SP)    ;DECIMAL
116 043040  104405                        TYPDS
117 043042  000405                        BR      19$
118 043044  013446                17$:    MOV     @(R4)+,-(SP)
119 043046  104402                        TYPOC
120 043050  000402                        BR      19$
121 043052  013446                18$:    MOV     @(R4)+,-(SP)
122 043054  104406                        TYPBN
123 043056  005714                19$:    TST     (R4)            ;MORE DATA??
124 043060  001403                        BEQ     20$             ;NO!!
125 043062  104401  043145    .           TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
126 043066  000760                        BR      16$             ;AND CONTINUE
127 043070  104401  001217        20$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
128 043074  000742                        BR      14$             ;BEFORE NEXT HEADER
129 043076  104415                21$:    RESREG
130 043100  000207                        RTS     PC
131
132 043102  000000                TSTNMB: .WORD   0               ;TEST NUMBER
133 043104  000000                ERRNMB: .WORD   0               ;ERROR NUMBER
134 043106  000000                BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
135 043110     000                BOTFLG: .BYTE   0               ;BOT FLAG
136 043111     000                CHRCNT: .BYTE   0               ;CHARACTER COUNT
137
138 043112     125     116     111  ERTY00: .ASCIZ  @UNIT#@
139 043120     054     040     124  ERTY01: .ASCIZ  @, TEST#@
140 043130     054     040     105  ERTY02: .ASCIZ  @, ERR#@
141 043137     054     040     120  ERTY03: .ASCIZ  @, PC=@
142 043145     040     040     000  ERTY04: .ASCIZ  @  @
143                                .EVEN
144
```

```
                        .SBTTL  TEST PREPARATION MODULE
                        ;
                        ; THIS MODULE PREPARES THE SUBSYSTEM FOR THE EXECUTION OF A TEST,
                        ; REPORTING AN ERROR TO THE USER IF AN ERROR IS  DETECTED.  THE USER
                        ; SPECIFIES TASKS TO BE PERFORMED, WHICH THE MODULE EXECUTES
                        ; USING SUBROUTINES.
                        ;
                        ;CALL:
                        ;       JSR     PC,TSTPRP
                        ;       .WORD   NNNNNN          TASK/VERIFY DESCRIPTOR
                        ;       BR      ??              RETURN HERE IF NO ERROR
                        ;       NOP                     RETURN HERE IF ERROR
                        ;       ERROR                   ERROR DEFINED BY MODULE
                        ;
                        ;TASK/VERIFY DESCRIPTOR
                        ;       BIT 15 = 1      SELECT DEVICE AND VERIFY DEVICE IS AVAILABLE
                        ;       ----------
                        ;       BIT 14 = 1      CLEAR CONTROLLER AND SELECT DEVICE
                        ;       BIT 13          (RESERVED FOR DRIVE CLEAR)
                        ;       BIT 12 = 1      PACK ACKNOWLEDGE IF VOLUME NOT VALID
                        ;       ----------
                        ;       BIT 11 = 1      RECALIBRATE IF POSITIONING IN PROGRESS
                        ;       BIT 10
                        ;       BIT 9
                        ;       ----------
                        ;       BIT 8
                        ;       BIT 7
                        ;       BIT 6  = 1      VERIFY CONTROLLER CLEAR OPERATION
                        ;       ----------
                        ;       BIT 5           (RESERVED FOR DRIVE CLEAR)
                        ;       BIT 4  = 1      VERIFY PACK ACKNOWLEDGE
                        ;       BIT 3  = 1      VERIFY RECALIBRATION
                        ;       ----------
                        ;       BIT 2
                        ;       BIT 1
                        ;       BIT 0
                        ;
043150                  TSTPRP:
                        ;
                        ;STORE TASK DESCRIPTOR AND CLEAR USER'S ERROR CALL
043150  017637  000000  044064          MOV     @(SP),500$      ;STORE DESCRIPTOR
043156  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USERS ERROR CALL
043162  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR CALL
043166  162716  000004                  SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
                        ;
043172  004737  044070                  JSR     PC,GETSTS       ;SETUP TO READ ALL REGISTERS
043176  004737  044154                  JSR     PC,GET          ;GET RMER2
043202  000411                          BR      15$             ;BR IF NO ERROR DETECTED
043204  000401                          BR      10$             ;GET OVER ERROR NUMBER
043206  000000                          .WORD   0               ;ERROR DEFINED BY GET SUBROUTINE
043210  062716  000004          10$:    ADD     #4,(SP)         ;XFER ERROR TO USER AND
043214  113776  043206  000000          MOVB    10$-2,@(SP)     ;GET ERROR NUMBER.
043222  000137  044054                  JMP     400$
                        ;
043226  013737  001376  044066  15$:    MOV     RMER21,505$     ;GET RMER2 AND SAVE FOR LATER
                        ;
                        ;;*****************************************************************
```

```
                                        ;SELECT DEVICE AND VERIFY DEVICE AVAILABLE IF BIT 15 SET IN TASK
043234  005737  044064                          TST     500$            ;SELECT DEVICE??
043240  100014                                  BPL     30$             ;NO!!

043242  004737  052056                          JSR     PC,DEVSEL       ;GO SELECT DEVICE
043246  000411                                  BR      30$             ;NO ERROR - CONTINUE
043250  000401                                  BR      20$
043252  000000                                  .WORD   0               ;ERROR NUMBER FROM DEVSEL
043254  062716  000004                20$:      ADD     #4,(SP)         ;TRANSFER ERROR TO USER
043260  113776  043252  000000                  MOVB    20$-2,@(SP)
043266  000137  044054                          JMP     400$

                                        ;;**************************************************************
                                        ;CLEAR CONTROLLER IF BIT 14 IS SET IN TASK
043272                                  30$:
043272  032737  040000  044064                  BIT     #BIT14,500$     ;CLEAR CONTROLLER??
043300  001451                                  BEQ     120$            ;NO.!

043302  004737  053530                          JSR     PC,CNTCLR       ;GO CLEAR CONTROLLER
043306  000411                                  BR      60$             ;CONTINUE - NO ERROR
043310  000401                                  BR      50$
043312  000000                          40$:    .WORD   0               ;ERROR NUMBER FROM CNTCLR
043314  062716  000004                50$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
043320  113776  043312  000000                  MOVB    40$,@(SP)
043326  000137  044054                          JMP     400$

                                        ;;**************************************************************
                                        ;VERIFY CONTROLLER CLEAR IF BIT6 SET IN TASK
043332                                  60$:
043332  032737  000100  044064                  BIT     #BIT6,500$      ;VERIFY??
043340  001431                                  BEQ     120$            ;NO.!

043342  004737  044154                          JSR     PC,GET          ;GO GET STATUS
043346  000411                                  BR      90$             ;NO ERROR GETTING STATUS
043350  000401                                  BR      80$
043352  000000                          70$:    .WORD   0               ;ERROR FROM GETTING STATUS
043354  062716  000004                80$:    ADD     #4,(SP)         ;TRANSFER ERROR TO USER
043360  113776  043352  000000                  MOVB    70$,@(SP)
043366  000137  044054                          JMP     400$

043372  004737  053646                90$:    JSR     PC,CLRSTS       ;GO VERIFY STATUS CLEAR
043376  000412                                  BR      120$            ;NO ERROR IN CLEAR
043400  000401                                  BR      110$
043402  000000                         100$:    .WORD   0               ;ERROR IN STATUS CLEAR
043404  005726                         110$:    TST     (SP)+           ;STRIP RETURN ADDRESS TO
043406  062716  000004                         ADD     #4,(SP)         ;SUBROUTINE AND TRANSFER
043412  113776  043402  000000                  MOVB    100$,@(SP)      ;ERROR TO USER
043420  000137  044054                          JMP     400$

                                        ;;**************************************************************
                                        ;EXECUTE PACK ACKNOWLEDGE IF BIT12 SET IN TASK AND VOLUME IS
                                        ;NOT VALID
043424                                  120$:
043424  032737  010000  044064                  BIT     #BIT12,500$     ;PACK ACKNOWLEDGE??
043432  001501                                  BEQ     240$            ;NO.!

043434  004737  044154                          JSR     PC,GET
```

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

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

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

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

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

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

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

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

                                                JSR     PC,GET          ;GO GET RMDS
                                                BR      270$            ;NO ERROR GETTING RMDS
                                                BR      260$
                                        250$:   .WORD   0               ;ERROR FROM GET SUB
                                        260$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
```

```
    172 043664  113776  043656  000000          MOVB    250$,@(SP)
    173 043672  000470                          BR      400$
    174
    175 043674  032737  040000  044066  270$:   BIT     #SKI,505$       ;WAS SKI SET ?
    176 043702  001004                          BNE     280$            ;YES, GO RECALIBRATE
    177 043704  032737  020000  001346          BIT     #PIP,RMDSI      ;IS PIP ACTIVE??
    178 043712  001462                          BEQ     410$            ;NO.!
    179
    180 043714  012737  000007  001410  280$:   MOV     #RECAL!GO,RMCS10        ;LOAD RECALIBRATE COMMAND
    181 043722  112737  000000  001543          MOVB    #RMCS1,PUTINX   ;AND REGISTER INDEX
    182 043730  112737  000200  001544          MOVB    #200,PUTINX+1
    183 043736  004737  044424                  JSR     PC,PUT          ;GO ISSUE RECALIBRATE
    184 043742  000410                          BR      300$            ;NO ERROR
    185 043744  000401                          BR      290$
    186 043746  000000                          .WORD   0               ;ERROR IN REGISTER TRANSFER
    187 043750  062716  000004          290$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
    188 043754  113776  043746  000000          MOVB    290$-2,@(SP)
    189 043762  000434                          BR      400$
    190
    191 043764  004737  044766          300$:   JSR     PC,TIMOUT       ;WAIT FOR COMPLETION
    192
    193                                 ;;*****************************************************************
    194                                 ;VERIFY RECALIBRATE IF BIT 3 SET IN TASK
    195 043770  032737  000010  044064          BIT     #BIT3,500$      ;VERIFY RECALIBRATE??
    196 043776  001430                          BEQ     410$            ;NO!!
    197
    198 044000  004737  044154                  JSR     PC,GET          ;GO GET STATUS
    199 044004  000410                          BR      330$            ;NO ERROR GETTING STATUS
    200 044006  000401                          BR      320$
    201 044010  000000                  310$:   .WORD   0               ;ERROR FROM GET
    202 044012  062716  000004          320$:   ADD     #4,(SP)         ;TRANSFER ERROR TO USER
    203 044016  113776  044010  000000          MOVB    310$,@(SP)
    204 044024  000413                          BR      400$
    205
    206 044026  004737  055322          330$:   JSR     PC,RCLSTS       ;GO CHE^K RECALIBRATE
    207 044032  000412                          BR      410$            ;NO ERROR DURING RECALIBRATE
    208 044034  000401                          BR      350$
    209 044036  000000                  340$:   .WORD   0               ;ERROR DURING RECALIBRATE
    210 044040  005726                  350$:   TST     (SP)+           ;STRIP RETURN TO SUB AND
    211 044042  062716  000004                  ADD     #4,(SP)         ;TRANSFER ERROR TO USER
    212 044046  113776  044036  000000          MOVB    340$,@(SP)
    213 044054  162716  000002          400$:   SUB     #2,(SP)         ;MOVE SP BACK BEFORE ERROR
    214 044060  000240                  410$:   NOP
    215 044062  000207                          RTS     PC              ;RETURN TO USER
    216
    217 044064  000000                  500$:   .WORD   0               ;TASK/VERIFY DESCRIPTOR
    218 044066  000000                  505$:   .WORD   0               ;CONTAINS RMER2
    219
```

```
                        .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

044070                  GETSTS:
044070  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044072  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044074  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044076  012700  001514          MOV     #GETINX,R0      ;R0 = ADDRESS OF INDEX TABLE
044102  012701  001404          MOV     #RMEC2I+2,R1    ;R1 = ADDRESS OF GET BUFFER
044106  012702  000046          MOV     #RMEC2,R2       ;R2 = REGISTER INDE
044112  110220          2$:     MOVB    R2,(R0)+        ;WRITE REGISTER INDEX IN TABLE
044114  005041                  CLR     -(R1)           ;CLEAR CORRESPONDING LOCATION
044116  162702  000002  3$:     SUB     #2,R2           ;DECREMENT TO NEXT INDEX
044122  100405                  BMI     4$              ;BRANCH OUT IF DONE
044124  022702  000022          CMP     #RMDB,R2        ;DONT WRITE RMDB INDEX
044130  001370                  BNE     2$
044132  005041                  CLR     -(R1)
044134  000770                  BR      3$
044136  112720  000200  4$:     MOVB    #200,(R0)+      ;WRITE TERMINATOR
044142  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
044144  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
044146  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
044150  000240                  NOP
044152  000207                  RTS     PC

30
```

```
                        .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

000240                  GET:    NOP
062716  000004                ADD     #4,(SP)         ;CLEAR ERROR NUMBER IN USER'S
105076  000000                CLRB    @(SP)           ;ERROR CALL
162716  000004                SUB     #4,(SP)
010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
013700  001276                  MOV     $BASE,R0
012702  001334                  MOV     #GETBUF,R2
012704  001514                  MOV     #GETINX,R4
012737  044336  000004          MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
012737  000300  000006          MOV     #PR6,ERRVEC+2
016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
001007                          BNE     3$              ;YES!!
062766  000004  000016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN USER'S
112776  000112  000016          MOVB    #112,@16(SP)    ;ERROR CALL
000423                          BR      7$
105714                  3$:     TSTB    (R4)            ;DONE??
100433                          BMI     9$              ;YES!!
111401                          MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
042701  177700                  BIC     #^C1DXMSK,R1    ;CLEAR ANY SIGN EXTENSION
```

```
                                                ADD     R0,R1
                                                MOVB    (R4)+,R3        ;R3 = STORAGE ADDRESS FOR REGISTER
                                                BIC     #^C1DXMSK,R3    ;CLEAR ANY SIGN EXTENSION
                                                ADD     R2,R3
                                                MOV     (R1),(R3)       ;READ REGISTER
                                                BR      3$

                                        5$:     CMP     (SP)+,(SP)+     ;RESTORE STACK
                                                ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
                                                MOVB    #7,@16(SP)      ;USER'S ERROR CALL
                                        7$:     SUB     #2,16(SP)
                                        8$:     TSTB    (R4)            ;DONE CLEARING??
                                                BMI     9$              ;YES!!
                                                CLR     R3              ;CLEAR REMAINING STORAGE
                                                MOVB    (R4)+,R3        ;LOCATIONS
                                                ADD     R2,R3
                                                CLR     (R3)
                                                BR      8$

                                        9$:
                                                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
```

```
                                .SBTTL  PUT SUBROUTINE

                                ;THIS SUBROUTINE WRITES THE REGISTERS WHICH ARE LISTED IN THE
                                ;"PUT INDEX TABLE" WITH  THE CONTENTS OF THE CORRESPONDING
                                ;LOCATION IN THE "PUT REGISTER BUFFER".  THE NUMBER OF
                                ;REGISTERS WRITTEN IS VARIABLE; THE INDEX TABLE MUST
                                ;BE TERMINATED WITH A CONTROL BYTE (200) WHICH SHOULD
                                ;FOLLOW THE LAST ENTRY.
                                ;
                                ;SUBROUTINE CALL:
                                ;
                                ;(1)     "PUT INDEX TABLE" HAS BEEN LOADED WITH INDEX VALUES
                                ;        OF REGISTERS TO BE WRITTEN.
                                ;(2)     "PUT REGISTER BUFFER" CONTAINS CONTENTS OF EACH
                                ;        REGISTER TO BE WRITTEN.
                                ;(3)     JSR     PC,PUT
                                ;        BR      ???             RETURN HERE IF NO ERROR FOUND
                                ;        NOP                     RETURN HERE IF ANY ERROR FOUND
                                ;        ERROR                   SUB DEFINES ERROR NUMBER
                                ;        ???
                                ;
                                ;R0 = REGISTER BASE ADDRESS
                                ;R1 = REGISTER ADDRESS
                                ;R2 = BUFFER BASE ADDRESS
                                ;R3 = BUFFER ADDRESS
                                ;R4 = POINTER TO REGISTER INDEX

044424  000240          PUT:    NOP
044426  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044430  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044432  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044434  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
044436  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
044440  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
044444  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
044450  013700  001276          MOV     $BASE,R0
044454  012702  001410          MOV     #PUTBUF,R2
044460  012704  001543          MOV     #PUTINX,R4
044464  012737  044574  000004  MOV     #5$,ERRVEC      ;SETUP FOR TIMEOUT
044472  012737  000300  000006  MOV     #PR6,ERRVEC+2
044500  016037  000010  001174  1$:     MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS
044506  016037  000000  001176          MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
044514  032737  004000  001176          BIT     #DVA,$TMP1      ;DEVICE AVAILABLE??
044522  001007                  BNE     3$              ;YES!!
044524  062766  000004  000016  ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
044532  112776  000112  000016  MOVB    #112,@16(SP)    ;USER'S ERROR CALL
044540  000424                  BR      7$
044542  105714          3$:     TSTB    (R4)            ;DONE??
044544  100425                  BMI     9$              ;YES!!
044546  111401                  MOVB    (R4),R1         ;R1 = REGISTER ADDRESS
044550  042701  177700          BIC     #^C1DXMSK,R1    ;CLEAR ANY SIGN EXTENSION
044554  060001                  ADD     R0,R1
044556  111403                  MOVB    (R4),R3         ;R3 = STORAGE ADDRESS
044560  042703  177700          BIC     #^C1DXMSK,R3    ;CLEAR ANY SIGN EXTENSION
044564  060203                  ADD     R2,R3
044566  011311                  MOV     (R3),(R1)       ;WRITE REGISTER
044570  105724                  TSTB    (R4)+           ;ADJUST REGISTER POINTER
```

```
     52 044572  000763                          BR      3$
     53
     54 044574  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
     55 044576  062766  000004  C00016          ADD     #4,16(SP)       ;WRITE ERROR NUMBER IN
     56 044604  112776  000007  000016          MOVB    #7,@16(SP)      ;USER'S ERROR CALL
     57 044612  162766  000002  000016  7$:     SUB     #2,16(SP)
     58
     59 044620                          9$:
        044620  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
        044624  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
        044630  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        044632  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        044634  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        044636  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        044640  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
     60 044642  000207                          RTS     PC              ;RETURN
     61
```

```
                                .SBTTL  SIZE CLOCK SUBROUTINE

044644                          SIZCLK:
044644  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
044650  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
044654  012737  044712  000004          MOV     #1$,ERRVEC      ;SET UP FOR BUS TIMEOUT
044662  012737  000300  000006          MOV     #PR6,ERRVEC+2
044670  012737  177546  001510          MOV     #177546,CLKADR  ;LOAD ADDRESSES FOR KW11-L
044676  012737  000100  001512          MOV     #100,CLKVCT
044704  005777  134600                  TST     @CLKADR         ;TEST FOR KW11-L PRESENT
044710  000421                          BR      3$              ;YES - KW11-L IS PRESENT
044712  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP
044714  012737  044744  000004          MOV     #2$,ERRVEC      ;SET UP FOR BUS TIMEOUT
044722  012737  172540  001510          MOV     #172540,CLKADR  ;LOAD ADDRESSES FOR KW11-P CLOCK
044730  012737  000104  001512          MOV     #104,CLKVCT
044736  005777  134546                  TST     @CLKADR         ;TEST FOR KW11-P PRESENT
044742  000404                          BR      3$              ;YES - KW11-P IS PRESENT
044744  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP
044746  062766  000002  000004          ADD     #2,4(SP)        ;MOVE RETURN TO ERROP
044754                          3$:
044754  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
044760  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
044764  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
044766                  TIMOUT:
044766  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044770  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044772  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044774  013746  000004                MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
045000  013746  000006                MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
045004  012737  045106  000004        MOV     #4$,ERRVEC      ;SETUP FOR BUS TIMEOUT - 04 TRAP
045012  012737  000300  000006        MOV     #PR6,ERRVEC+2
045020  013700  001276                MOV     $BASE,R0        ;R0=BASE ADDRESS
045024  013701  001510                MOV     CLKADR,R1       ;R1=CLOCK ADDRESS
045030  012702  000036                MOV     #30.,R2         ;R2=NUMBER OF CLOCK CYCLES
045034  020127  172540        1$:     CMP     R1,#172540      ;KW11-P CLOCK??
045040  001003                        BNE     2$              ;NO!!
045042  012761  000001  000002        MOV     #1,2(R1)        ;SET COUNTER
045050  012711  000005        2$:     MOV     #BIT2!BIT0,(R1) ;START COUNTER

045054  016046  000000        3$:     MOV     RMCS1(R0),-(SP) ;GET STATUS
045060  042716  177576                BIC     #^C<RDY!GO>,(SP)
045064  022726  000200                CMP     #RDY,(SP)+      ;RDY=1,GO=0??
045070  001420                        BEQ     5$              ;YES!!
045072  032711  000200                BIT     #BIT7,(R1)      ;TIMER DONE??
045076  001766                        BEQ     3$              ;NO.!
045100  005302                        DEC     R2              ;DEC NUMBER OF CYCLES
045102  001354                        BNE     1$              ;CONTINUE IF NOT DONE
045104  000412                        BR      5$              ;''RDY'' DID NOT SET OR ''GO'' DID NOT RESET
                                                            ;WITHIN 500 MSEC AFTER THE COMMAND WAS ISSUED.
045106  022626                4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
045110  062766  000004  000012        ADD     #4,12(SP)       ;MOVE SP TO USER'S CALL
045116  112776  000007  000012        MOVB    #7,@12(SP)      ;WRITE ERROR NUMBER
045124  162766  000002  000012        SUB     #2,12(SP)

045132                  5$:
045132  012637  000006                MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
045136  012637  000004                MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
045142  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
045144  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
045146  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
045150  000207                        RTS     PC              ;RETURN TO USER
```

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

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

045152                  PRIERR:

                        ;CLEAR USER'S ERROR CALL
045152  062716  000004          ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
045156  105076  000000          CLRB    @(SP)           ;CLEAR ERROR NUMBER
045162  162716  000004          SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

                        ;REPORT AN ERROR IF THE WRONG UNIT IS SELECTED
045166  013737  001344  001142  MOV     RMCS21,$BDDAT   ;CORRECT UNIT SELECTED??
045174  042737  177770  001142  BIC     #^CUNTMSK,$BDDAT
045202  013737  001234  001140  MOV     $UNIT,$GDDAT    ;GOOD DATA FOR TYPEOUT
045210  042737  177770  001140  BIC     #^CUNTMSK,$GDDAT
045216  123737  001140  001142  CMPB    $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
                                                        ;DRIVE NUMBERS
045224  001415                  BEQ     1$              ;YES!!
045226  062716  000004          ADD     #4,(SP)
045232  112776  000001  000000  MOVB    #1,@(SP)        ;ERROR 1
```

```
 58 045240  162716  000002                    SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
 59 045244  004736                            JSR     PC,@(SP)+        ;REPORT WRONG UNIT SELECTED
 60 045246  162716  000010                    SUB     #10,(SP)         ;RESTORE (SP)
 61 045252  000240                            NOP
 62 045254  000137  045774                    JMP     10$              ;SKIP OTHER CHECKS
 63 045260                              1$:
 64
 65                                     ;REPORT AN ERROR IF THE DEVICE IS NOT AVAILABLE OR IF
 66                                     ;THE DEVICE IS NONEXISTANT
 67 045260  032737  004000  001334            BIT     #DVA,RMCS1I      ;DEVICE AVAILABLE??
 68 045266  001045                            BNE     5$               ;YES!!
 69 045270  013737  001334  001140            MOV     RMCS1I,$GDDAT    ;EXPECTED STATUS
 70 045276  052737  004000  001140            BIS     #DVA,$GDDAT
 71 045304  013737  001334  001142            MOV     RMCS1I,$BDDAT    ;RECEIVED STATUS
 72 045312  062716  000004                    ADD     #4,(SP)
 73 045316  112776  000002  000000            MOVB    #2,@(SP)         ;ERROR #2
 74 045324  032737  010000  001344            BIT     #NED,RMCS2I      ;WAS NED SET??
 75 045332  001414                            BEQ     2$               ;NO!!
 76 045334  013737  001344  001140            MOV     RMCS2I,$GDDAT    ;EXPECTED STATUS
 77 045342  013737  001344  001142            MOV     RMCS2I,$BDDAT    ;RECEIVED STATUS
 78 045350  042737  010000  001140            BIC     #NED,$GDDAT
 79 045356  112776  000003  000000            MOVB    #3,@(SP)         ;YES - CHANGE ERROR NUMBER
 80 045364  162716  000002              2$:   SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
 81 045370  004736                            JSR     PC,@(SP)+        ;REPORT DEVICE NOT AVAILABLE
 82 045372  162716  000010                    SUB     #10,(SP)         ;RESTORE (SP)
 83 045376  000240                            NOP
 84 045400  000575                            BR      10$              ;SKIP OTHER CHECKS
 85 045402                              5$:
 86
 87                                     ;REPORT AN ERROR IF MASSBUS CONTROLLER IS NOT READY
 88 045402  032737  000200  001334            BIT     #RDY,RMCS1I      ;CONTROLLER READY??
 89 045410  001030                            BNE     7$               ;YES!!
 90 045412  013737  001334  001140            MOV     RMCS1I,$GDDAT    ;EXPECTED STATUS
 91 045420  052737  000200  001140            BIS     #RDY,$GDDAT
 92 045426  042737  160001  001140            BIC     #SC!TRE!MCPE!GO,$GDDAT
 93 045434  013737  001334  001142            MOV     RMCS1I,$BDDAT    ;RECEIVED STATUS
 94 045442  062716  000004                    ADD     #4,(SP)
 95 045446  112776  000004  000000            MOVB    #4,@(SP)         ;ERROR #4
 96 045454  162716  000002                    SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
 97 045460  004736                            JSR     PC,@(SP)+        ;REPORT CONTROLLER NOT READY
 98 045462  162716  000010                    SUB     #10,(SP)         ;RESTORE (SP)
 99 045466  000240                            NOP
100 045470  000541                            BR      10$              ;SKIP OTHER CHECKS
101 045472                              7$:
102
103                                     ;REPORT AN ERROR IF GO IS NOT ZERO AND DRY IS NOT ONE
104 045472  032737  000001  001334            BIT     #GO,RMCS1I       ;GO RESET??
105 045500  001431                            BEQ     8$               ;YES!!
106 045502  032737  000200  001346            BIT     #DRY,RMDSI       ;DRIVE READY??
107 045510  001025                            BNE     8$               ;YES!!
108 045512  013737  001334  001140            MOV     RMCS1I,$GDDAT    ;EXPECTED STATUS
109 045520  042737  160001  001140            BIC     #SC!TRE!MCPE!GO,$GDDAT
110 045526  013737  001334  001142            MOV     RMCS1I,$BDDAT    ;RECEIVED STATUS
111 045534  062716  000004                    ADD     #4,(SP)
112 045540  112776  000005  000000            MOVB    #5,@(SP)         ;ERROR #5
113 045546  162716  000002                    SUB     #2,(SP)          ;MOVE SP TO RETURN FOR ERROR
114 045552  004736                            JSR     PC,@(SP)+        ;REPORT DRIVE NOT READY
```

```
                                        SUB     #10,(SP)        ;RESTORE (SP)
                                        NOP
                                        BR      10$
                                8$:

                                ;REPORT AN ERROR IF THE RH CONTROLLER DETECTED BAD
                                ;PARITY ON THE MASSBUS CONTROL BUS
                                        BIT     #MCPE,RMCS1I    ;PARITY ERROR ??
                                        BEQ     9$              ;NO!!
                                        MOV     RMCS1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #SC!TRE!MCPE!GO,$GDDAT
                                        MOV     RMCS1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
                                        MOVB    #13,@(SP)       ;ERROR #47
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                        SUB     #10,(SP)        ;RESTORE STACK
                                        NOP
                                        BR      10$
                                9$:

                                ;REPORT AN ERROR IF DETECTED A CONTROL BUS PARITY ERROR
                                        BIT     #PAR,RMER1I     ;WAS THERE A PARITY ERROR??
                                        BEQ     11$             ;NO!!
                                        BIT     #DPE,RMER2I     ;WAS IT THE CONTROL BUS??
                                        BNE     11$             ;NOT SURE!!
                                        BIT     #PAR,RMER1O     ;DID TEST SET PAR ??
                                        BEQ     93$             ;NO!!
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     #PUTINX,R0      ;R0 POINTS TO INDEX TABLE
                                91$:    CMPB    #RMER1,(R0)     ;SEARCH TABLE FOR RMER1
                                        BNE     92$
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        BR      11$             ;PAR WAS SET BY TEST
                                92$:    TSTB    (R0)+           ;END OF TABLE??
                                        BPL     91$             ;NO!!
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                93$:    MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
                                        BIC     #PAR,$GDDAT
                                        MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
                                        ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #50,@(SP)       ;WRITE THE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT THE ERROR
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
                                        NOP
                                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 THE
                                ;INPUT REGISTER BUFFER.

046004                          SECERR:

                                ;;*****************************************************************
                                ;STORE FUNCTION CODE AND CLEAR USER'S ERROR NUMBER
046004  013737  001410  051640          MOV     RMCS10,515$     ;STORE FUNCTION CODE
046012  042737  177701  051640          BIC     #^C<F0!F1!F2!F3!F4>,515$
046020  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
046024  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
046030  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN

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

                                ;REPORT ERROR IF DRIVE IS NOT READY, I.E., IF DRY = 0
046034  032737  000200  001346          BIT     #DRY,RMDSI      ;DRIVE READY??
046042  001024                          BNE     5$              ;YES!!
046044  013737  001346  001142          MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
046052  042737  177577  001142          BIC     #^CDRY,$BDDAT
046060  012737  000200  001140          MOV     #DRY,$GDDAT     ;GOOD DATA FOR TYPEOUT
046066  062716  000004                  ADD     #4,(SP)
046072  112776  000010  000000          MOVB    #10,@(SP)       ;ERROR NUMBER
046100  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
046104  004736                          JSR     PC,@(SP)+       ;REPORT NOT READY
046106  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP) TO ERROR N
046112  000240                          NOP

                                ;REPORT ERROR IF GO BIT IS NOT RESET
046114  032737  000001  001334  5$:     BIT     #GO,RMCS11      ;GO BIT RESET??
046122  001423                          BEQ     10$             ;YES!!
046124  013737  001334  001142          MOV     RMCS11,$BDDAT   ;BAD DATA FOR TYPEOUT
046132  042737  177776  001142          BIC     #^CGO,$BDDAT
046140  005037  001140                  CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
046144  062716  000004                  ADD     #4,(SP)
046150  112776  000011  000000          MOVB    #11,@(SP)       ;ERROR NUMBER
```

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

                                ;REPORT ERROR IF FUNCTION CODE READ FROM DEVICE IS NOT CORRECT
                                10$:    MOV     RMCS11,$BDDAT   ;IS FUNCTION CODE CORRECT??
                                        BIC     #^C76,$BDDAT
                                        MOV     515$,$GDDAT     ;EXPECTED FUNCTION CODE
                                        CMP     $BDDAT,$GDDAT
                                        BEQ     15$             ;YES!!
                                        ADD     #4,(SP)
                                        MOVB    #12,@(SP)       ;ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT WRONG FUNCTION CODE
                                        SUB     #10,(SP)                ;RESTORE (SP)
                                        NOP
                                15$:
                                ;REPORT AN ERROR IF COMPOSITE ERROR IS SET AND NO OTHER
                                ;ERRORS ARE SET, OR IF COMPOSITE ERROR IS NOT SET AND
                                ;OTHER ERRORS ARE SET
                                        CLR     $GDDAT          ;EXPECT "ERR" = 0
                                        TST     RMER11          ;IS RMER1 = 0??
                                        BNE     20$             ;NO.!
                                        TST     RMER21          ;IS RMERZ = 0??
                                        BEQ     25$             ;YES!!
                                20$:    BIS     #ERR,$GDDAT     ;"ERR" SOULD BE SET
                                25$:    MOV     RMDS1,$BDDAT
                                        BIC     #^CERR,$BDDAT
                                        CMP     $GDDAT,$BDDAT   ;IS "ERR" OK??
                                        BEQ     30$             ;YES!!
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
                                        MOVB    #47,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT INVALID COMP ERROR
                                        SUB     #10,(SP)

                                ;REPORT AN ERROR IF "TRE" IS SET AND NONE OF THE BITS WHICH SET
                                ;TRE IS SET, OR IF TRE IS NOT SET AND ONE OR MORE BITS WHICH
                                ;SET TRE IS SET
                                30$:    CLR     $GDDAT          ;EXPECT "TRE" = 0
                                        MOV     RMCS21,-(SP)    ;WAS DLT, WCE, UPE, NED, NEM
                                        BIC     #377,(SP)+      ;PGE, MXF OR MDPE SET
                                        BNE     35$             ;YES!!
                                        BIT     #ERR,RMDSI      ;WAS EXCEPTION RECEIVED??
                                        BEQ     40$             ;NO!!
                                        CMP     #SEARCH,515$    ;WAS DATA TRANSFERRED??
                                        BHIS    40$             ;NO!!
                                35$:    BIS     #TRE,$GDDAT     ;"TRE" SHOULD BE SET
                                40$:    MOV     RMCS11,$BDDAT   ;BAD DATA FOR TYPEOUT
                                        BIC     #^CTRE,$BDDAT
                                        CMP     $GDDAT,$BDDAT   ;IS "TRE" OK??
                                        BEQ     45$             ;YES!!
                                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                        MOVB    #14,@(S         ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                        JSR     PC,@(SP)+       ;REPORT TRE ERROR
```

```
046460  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
046464  000240                          NOP
046466                          45$:

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

                                ;GET AND STORE THE ENTRY FROM THE FUNCTION CODE TABLE
046466  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
046470  013700  051640                  MOV     515$,R0         ;GET FUNCTION CODE
046474  016037  067046  051632          MOV     FNCDTB(R0),500$ ;STORE ENTRY
046502  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.
046504  013737  051632  001140          MOV     500$,$GDDAT     ;GET EXPECTED ATA STATUS
046512  032737  040000  001346          BIT     #ERR,RMDSI      ;IS COMPOSITE ERROR SET ??
046520  001403                          BEQ     50$             ;NO !!
046522  052737  100000  001140          BIS     #ATA,$GDDAT     ;EXPECT AN ATTENTION
046530  042737  077777  001140  50$:    BIC     #^CATA,$GDDAT   ;STRIP DONT CARES
046536  013737  001346  001142          MOV     RMDSI,$BDDAT    ;GET RECEIVED ATA
046544  042737  077777  001142          BIC     #^CATA,$BDDAT   ;STRIP DONT CARES
046552  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ATA OK ??
046560  001413                          BEQ     55$             ;YES !!
046562  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
046566  112776  000006  000000          MOVB    #6,@(SP)        ;LOAD ERROR # IN CALL
046574  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
046600  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
046602  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
046606  000240                          NOP
046610                          55$:

                                ;REPORT ERROR IF ILF IS INCORRECT, I.E., IF ILF DOES NOT COMPARE
                                ;WITH FUNCTION CODE TABLE
046610  013737  051632  001140          MOV     500$,$GDDAT     ;GET EXPECTED ILF
046616  042737  177776  001140          BIC     #^CILF,$GDDAT   ;CLEAR ALL OTHER BITS
046624  013737  001350  001142          MOV     RMER1I,$BDDAT   ;GET RECEIVED ILF
046632  042737  177776  001142          BIC     #^CILF,$BDDAT   ;CLEAR ALL OTHER BITS
046640  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS ILF OK ??
046646  001412                          BEQ     60$             ;YES !!
046650  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
046654  112776  000254  000000          MOVB    #254,@(SP)      ;WRITE ERROR NUMBER IN CALL
046662  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
046666  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
046670  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
046674  005037  001140          60$:    CLR     $GDDAT          ;CLEAR EXPECTED STATUS

                                ;REPORT AN ERROR IF WCE IS SET AND SHOULD NOT BE SET
046700  013746  051632                  MOV     500$,-(SP)      ;GET WCE STATUS ENABLE
046704  052716  137777                  BIS     #^CWCE,(SP)     ;SET ALL OTHER BITS
046710  013737  001344  001142          MOV     RMCS2I,$BDDAT   ;RECEIVED STATUS
046716  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR WCE IF ENABLED
```

```
172 046722  001412                          BEQ     90$             ;BRANCH IF WCE OK
173 046724  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
174 046730  112776  000026  000000          MOVB    #26,@(SP)       ;WRITE ERROR NUMBER
175 046736  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
176 046742  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
177 046744  162716  000010                  SUB     #10,(SP)        ;RESTORE ERROR
178 046750                          90$:
179
180                                 ;REPORT ERROR IF OPI STATUS IS SET AND SHOULD NOT BE SET
181 046750  013746  051632                  MOV     500$,-(SP)      ;GET OPI STATUS ENABLE
182 046754  052716  157777                  BIS     #^COPI,(SP)     ;SET ALL OTHER BITS
183 046760  013737  001350  001142          MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
184 046766  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR OPI IF ENABLED
185 046772  001412                          BEQ     100$            ;BRANCH IF OPI OK
186 046774  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
187 047000  112776  000164  000000          MOVB    #164,@(SP)      ;WRITE ERROR NUMBER IN CALL
188 047006  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
189 047012  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
190 047014  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
191 047020                          100$:
192
193                                 ;REPORT ERROR IF IVC IS SET AND IS NOT ENABLED OR IF IVC IS
194                                 ;SET AND VV IS NOT RESET
195 047020  013746  051632                  MOV     500$,-(SP)      ;GET IVC STATUS ENABLE
196 047024  032737  000100  001346          BIT     #VV,RMDSI       ;IS VV SET
197 047032  001402                          BEQ     105$            ;NO !!
198 047034  042716  010000                  BIC     #IVC,(SP)       ;YES - IVC SHOULD BE 0
199 047040  052716  167777          105$:   BIS     #^CIVC,(SP)     ;SET ALL OTHER BITS
200 047044  013737  001376  001142          MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
201 047052  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR IVC IF ENABLED
202 047056  001412                          BEQ     110$            ;BRANCH IF IVC OK
203 047060  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
204 047064  112776  000165  000000          MOVB    #165,@(SP)      ;WRITE ERROR NUMBER IN CALL
205 047072  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
206 047076  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
207 047100  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
208 047104                          110$:
209
210                                 ;BIT 11 (WLE) OF THE FUNCTION CODE TABLE IS THE ENABLING BIT FOR
211                                 ; ALL WRITE ERRORS, I.E.,
212                                 ;       RMER1 - WLE, WCF
213                                 ;       RMER2 - DPE
214                                 ;       RMCS2 - UPE.
215                                 ;EACH OF THESE ERRORS IS CHECKED TO SEE IF AN ERROR IS SET WHEN THE
216                                 ;WRITE ERROR ENABLE BIT IS RESET.
217
218                                 ;REPORT AN ERROR IF WLE IS SET AND WRITE ERRORS ARE NOT ENABLED, OR IF
219                                 ;THE DRIVE IS NOT WRITE PROTECTED
220 047104  012746  177777                  MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
221 047110  032737  004000  051632          BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
222 047116  001404                          BEQ     115$            ;NO !!
223 047120  032737  004000  001346          BIT     #WRL,RMDSI      ;IS THE DRIVE WRITE PROTECTED ??
224 047126  001002                          BNE     120$            ;YES !!
225 047130  042716  004000          115$:   BIC     #WLE,(SP)       ;RESET WLE ENABLE
226 047134  013737  001350  001142  120$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
227 047142  042637  001142                  BIC     (SP)+,$BDDAT    ;CLEAR WLE IF ENABLED
228 047146  001412                          BEQ     125$            ;BRANCH IF WLE OK
```

```
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #23,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        125$:

                                        ;REPORT ERROR IF WCF IS SET AND WRITE ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ENABLED
                                                BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                                BNE     130$            ;YES !!
                                                BIC     #WCF,(SP)       ;DISABLE WCF ERROR
                                        130$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET WCF IF ENABLED
                                                BEQ     135$            ;BRANCH IF WCF OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #25,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        135$:

                                        ;REPORT ERROR IF DPE IS SET AND WRITE ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
                                                BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                                BNE     140$            ;YES !!
                                                BIC     #DPE,(SP)       ;RESET DPE ENABLE
                                        140$:   MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET DPE IF ENABLED
                                                BEQ     145$            ;BRANCH IF DPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #40,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR
                                                SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
                                        145$:

                                        ;REPORT AN ERROR IF UPE IS SET AND WRITE ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME WRITE ERRORS ARE ENABLED
                                                BIT     #WLE,500$       ;ARE WRITE ERRORS ENABLED ??
                                                BNE     150$            ;YES !!
                                                BIC     #UPE,(SP)       ;DISABLE UPE ERROR
                                        150$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET UPE IF ENABLED
                                                BEQ     155$            ;BRANCH IF UPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #24,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        155$:

                                        ;REPORT AN ERROR IF IAE IS SET AND IS NOT ENABLED
                                                MOV     500$,-(SP)      ;GET IAE ENABLE
                                                BIS     #^CIAE,(SP)     ;SET ALL OTHER BITS
                                                MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+, $BDDAT   ;CLEAR IAE IF ENABLED
```

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

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


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

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

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

```
                                        BIC     #MXF,(SP)       ;DISABLE MXF ERROR
                                185$:   MOV     RMCS2I $BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR MXF IF ENABLED
                                        BEQ     190$            ;BRANCH IF MXF IS OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #33,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                190$:
                                ;REPORT ERROR IF AOE IS SET AND DATA ERRORS ARE NOT ENABLED
                                        MOV     #-1,-(SP)       ;ASSUME DATA ERRORS ARE ENABLED
                                        BIT     #AOE,500$       ;ARE DATA ERRORS EANBLED ??
                                        BEQ     191$            ;NO !!
                                        BIT     #LBT,RMDS1      ;IS LBT ALSO SET ??
                                        BNE     195$            ;YES !!
                                191$:   BIC     #AOE,(SP)       ;DISABLE AOE
                                195$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                        BIC     (SP)+,$BDDAT    ;CLEAR AOE IF ENABLED
                                        BEQ     200$            ;BRANCH IF AOE IS OK
                                        ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        MOVB    #20,@(SP)       ;WRITE ERROR NUMBER
                                        SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                        JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                        SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                200$:

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

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

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

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

```
                                                BNE     215$            ;YES !!
                                                BIC     #HCE,(SP)       ;DISABLE HCE
                                        001142  215$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR HCE IF ENABLED
                                                BEQ     220$            ;BRANCH IF HCE IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000  MOVB    #36,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        220$:

                                        ;REPORT ERROR IF FER IS SET AND HEADER ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME FER IS ENABLED
                                        051632  BIT     #HCE,500$       ;ARE HEADER ERRORS ENABLED ??
                                                BNE     225$            ;YES !!
                                                BIC     #FER,(SP)       ;DISABLE FER
                                        001142  225$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;RESET FER IF ENABLED
                                                BEQ     230$            ;BRANCH IF FER OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000  MOVB    #37,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        230$:

                                        ;REPORT ERROR IF BSE IS SET AND HEADER ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ERRORS ENABLED
                                        051632  BIT     #HCE,500$       ;ARE THEY ENABLED ??
                                                BNE     235$            ;YES !!
                                                BIC     #BSE,(SP)       ;DISABLE BSE
                                        001142  235$:   MOV     RMER2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR BSE IF ENABLED
                                                BEQ     240$            ;BRANCH IF BSE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                        000000  MOVB    #354,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        240$:

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

                                        ;REPORT ERROR IF MDPE IS SET AND IS NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                        051632  BIT     #ECH,500$       ;ARE DATA FIELD ERRORS ENABLED ??
                                                BNE     245$            ;YES !!
                                                BIC     #MDPE,(SP)      ;DISBALE MDPE
                                        001142  245$:   MOV     RMCS2I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR MDPE IF ENABLED
```

```
                                                BEQ     250$            ;BRANCH IF MDPE OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #27,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        250$:
                                        ;REPORT ERROR IF DCK IS SET AND DATA FIELD ERRORS ARE NOT ENABLED
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                                BIT     #ECH,500$       ;ARE THEY ENABLED ??
                                                BNE     255$            ;YES !!
                                                BIC     #DCK,(SP)       ;DISABLE DCK
                                        255$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR DCK IF ENABLED
                                                BEQ     260$            ;BRANCH IF DCK IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #30,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        260$:
                                        ;REPORT ERROR IF ECH IS SET AND,
                                        ;       DATA FIELD ERRORS ARE NOT ENABLED, OR
                                        ;       ECI IS SET, OR
                                        ;       DCK IS NOT SET.
                                                MOV     #-1,-(SP)       ;ASSUME ENABLED
                                                BIT     #ECH,500$       ;ARE ERRORS ENABLED ??
                                                BEQ     265$            ;NO !!
                                                BIT     #ECI,RMOFI      ;IS ECI SET ??
                                                BNE     265$            ;YES !!
                                                BIT     #DCK,RMER1I     ;IS DCK ALSO SET ??
                                                BNE     270$            ;YES !!
                                        265$:   BIC     #ECH,(SP)       ;DISABLE ECH
                                        270$:   MOV     RMER1I,$BDDAT   ;GET RECEIVED STATUS
                                                BIC     (SP)+,$BDDAT    ;CLEAR ECH IF ENABLED
                                                BEQ     275$            ;BRANCH IF ECH IS OK
                                                ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
                                                MOVB    #31,@(SP)       ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
                                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
                                                SUB     #10,(SP)        ;MOVE SP TO NO ERROR
                                        275$:
```

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

050552  022737  000030  051640            CMP     #SEARCH,515$    ;WAS DATA TRANSFERRED ?
050560  103402                            BLO     280$            ;BR IF YES
050562  000137  051604                    JMP     355$            ;NO - EXIT

                                ;REPORT ERROR IF RMWC NOT ZERO AND TRE IS ZERO
050566  013737  001336  001142  280$:     MOV     RMWCI,$BDDAT    ;WORD COUNT ZERO??
050574  001421                            BEQ     285$            ;YES
050576  032737  040000  001334            BIT     #TRE,RMCS1I     ;TRANSFER ERROR DETECTED??
050604  001015                            BNE     285$            ;YES!!
050606  062716  000004                    ADD     #4,(SP)
050612  112776  000015  000000            MOVB    #15,@(SP)       ;ERROR NUMBER
050620  005037  001140                    CLR     $GDDAT          ;GOOD DATA FOR TYPEOUT
050624  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050630  004736                            JSR     PC,@(SP)+       ;REPORT WORD COUNT NOT ZERO
050632  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
050636  000240                            NOP

                                ;REPORT ERROR IF RMBA IS NOT CORRECT
050640  013737  001336  001140  285$:     MOV     RMWCI,$GDDAT    ;GET WORD COUNT AT END OF TRANSFER AND
050646  163737  001412  001140            SUB     RMWCO,$GDDAT    ;SUBTRACT STARTING WORD COUNT.
050654  006337  001140                    ASL     $GDDAT          ;* 2
050660  063737  001414  001140            ADD     RMBAO,$GDDAT    ;ADD STARTING BUS ADDRESS

050666  032737  000010  001344            BIT     #BAI,RMCS2I     ;WAS BUS ADDRESS INHIBIT (BAI) SET ??
050674  001403                            BEQ     290$            ;NO !!
050676  013737  001414  001140            MOV     RMBAO,$GDDAT    ;ADDRESS SHOULD NOT HAVE CHANGED

050704  023737  001140  001340  290$:     CMP     $GDDAT,RMBAI    ;BUS ADDRESS OK??
050712  001416                            BEQ     295$            ;YES..
050714  013737  001340  001142            MOV     RMBAI,$BDDAT    ;BAD DATA FOR TYPEOUT
050722  062716  000004                    ADD     #4,(SP)
050726  112776  000016  000000            MOVB    #16,@(SP)       ;ERROR NUMBER
050734  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
050740  004736                            JSR     PC,@(SP)+       ;REPORT UNEXPECTED ADDRESS
050742  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
050746  000240                            NOP

                                ;COMPUTE NUMBER OF SECTORS TRANSFERRED FROM WORD COUNT
050750  005046                  295$:     CLR     -(SP)           ;NUMBER OF SECTORS TRANSFERRED
050752  013746  001336                    MOV     RMWCI,-(SP)     ;GET WORD COUNT AT END OF TRANSFER AND
050756  163716  001412                    SUB     RMWCO,(SP)      ;SUBTRACT STARTING WORD COUNT.

050762  012746  000400                    MOV     #256.,-(SP)     ;ASSUME 256. WORDS PER SECTOR
050766  032737  000002  001410            BIT     #BIT1,RMCS1O    ;HEADER & DATA COMMAND ??
050774  001402                            BEQ     300$            ;NO !!
050776  062716  000002                    ADD     #2,(SP)         ;CHANGE TO 258. WORDS PER SECTOR

051002  005266  000004          300$:     INC     4(SP)           ;INCREMENT SECTOR COUNT
051006  161666  000002                    SUB     (SP),2(SP)      ;SUBTRACT ONE SECTOR'S WORTH
051012  003373                            BGT     300$            ;CONTINUE IF NOT DONE
051014  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK

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

```
                                        ;NUMBER OF SECTORS
051016  013737  001444  051632            MOV     RMDCO,500$      ;STORE ORIGINAL CYLINDER
051024  013737  001416  051634            MOV     RMDAO,505$      ;STORE ORIGINAL TRACK
051032  013737  001416  051636            MOV     RMDAO,510$      ;STORE ORIGINAL SECTOR
051040  013737  001332  051642            MOV     LSTRK,520$      ;STORE LAST TRACK,
051046  000337  051642                    SWAB    520$            ;GET TRACK ADDRESS TO LO BYTE AND
051052  005237  051642                    INC     520$            ;INCREMENT TO GET TOTL # OF TRACKS.

051056  042737  000377  051634            BIC     #^C<TADMSK>,505$        ;SAVE TRACK ADDRESS BITS AND
051064  000337  051634                    SWAB    505$            ;SWAP TRACK ADDRESS TO LOW BYTE.
051070  042737  177400  051636            BIC     #^C<SADMSK>,510$        ;SAVE SECTOR ADDRESS BITS
051076  062637  051636                    ADD     (SP)+,510$

051102  023727  051636  000040  310$:     CMP     510$,#32.       ;SECTOR OVEFLOWED??
051110  103406                            BLO     315$            ;NO!!
051112  005237  051634                    INC     505$            ;INCREMENT TRACK
051116  162737  000040  051636            SUB     #32.,510$       ;ADJUST SECTOR
051124  000766                            BR      310$            ;TRY AGAIN

051126  023737  051634  051642  315$:     CMP     505$,520$       ;TRACK OVERFLOWED??
051134  103407                            BLO     320$            ;NO!!
051136  005237  051632                    INC     500$            ;INCREMENT CYLINDER
051142  163737  051642  051634            SUB     520$,505$       ;ADJUST TRACK
051150  000766                            BR      315$            ;TRY AGAIN
051152  000240                            NOP

                                        ;REPORT ERROR IF "LBT" IS NOT CORRECT
051154                                  320$:
051154  005037  001140                    CLR     $GDDAT          ;SET GOOD DATA FOR LBT = 0
051160  023727  051632  001466            CMP     500$,#822.      ;SHOULD LBT BE SET??
051166  101407                            BLOS    325$            ;NO!!
051170  032737  002000  001350            BIT     #IAE,RMER1I     ;WAS IAE SET ??
051176  001003                            BNE     325$            ;YES - LBT SHOULD NOT BE SET
051200  012737  002000  001140            MOV     #LBT,$GDDAT     ;SET GOOD DATA FOR LBT = 1
051206  013737  001346  001142  325$:     MOV     RMDSI,$BDDAT    ;BAD DATA FOR TYPEOUT
051214  042737  175777  001142            BIC     #^CLBT,$BDDAT
051222  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;IS LBT CORRECT??
051230  001413                            BEQ     330$            ;YES!!
051232  062716  000004                    ADD     #4,(SP)
051236  112776  000017  000000            MOVB    #17,@(SP)       ;ERROR NUMBER
051244  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051250  004736                            JSR     PC,@(SP)+       ;REPORT LBT IS WRONG
051252  162716  000010                    SUB     #10,(SP)        ;RESTORE (SP)
051256  000240                            NOP

                                        ;REPORT ERROR IF "AOE" IS INCORRECT
051260  005037  001140                  330$:     CLR     $GDDAT          ;SET FOR AOE = 0
051264  032737  002000  001350            BIT     #IAE,RMER1I     ;WAS "IAE" DETECTED??
051272  001031                            BNE     340$            ;YES-"AOE" SHOULD BE ZERO
051274  023727  051632  001466            CMP     500$,#822.      ;SHOULD AOE BE SET??
051302  101425                            BLOS    340$            ;NO!!
051304  005737  051634                    TST     505$            ;MAYBE
051310  001012                            BNE     335$            ;YES
051312  005737  051636                    TST     510$
051316  001007                            BNE     335$            ;YES !!
051320  032737  000010  051640            BIT     #F2,515$        ;WAS THIS READ OR WRITE CHECK ??
051326  001413                            BEQ     340$            ;NO ..
```

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

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

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

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

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

```
    172 051642  000000                520$:   .WORD   0               ;# OF TRACKS FOR DEVICE UNDER TEST = LAST
    173                                                                 ;TRACK + 1 TRACK
    174
```

```
                                .SBTTL  COMPOSITE ERROR CHECK SUBROUTINE
                                
                                ;THIS SUBROUTINE CHECKS THE STORED CONTENTS OF RMER1 AND
                                ;RMER2 AFTER MASKING EACH REGISTER WORD WITH THE USER'S STATUS
                                ;MASKS AND REPORTS AN ERROR IF ANY BITS ARE LEFT ON AFTER
                                ;THE MASKS ARE APPLIED.
                                
                                ;CALL:
                                ;(1)    JSR     PC,CMPERRSTS
                                ;       .WORD                   MASK FOR ERROR REGISTER 1
                                ;       .WORD                   MASK FOR ERROR REGISTER 2
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS
                                
                                ;NOTE:  BITS TO BE MASKED SHOULD BE ONE; BITS TO BE TESTED SHOULD
                                ;BE ZERO
                                
051644                          CMPERRSTS:
                                
                                ;MASK AND STORE THE CONTENTS OF RMER1  AND RMER2
051644  013737  001350  001176          MOV     RMER1I,$TMP1    ;STORE RMER1 AT TEMP STORAGE
051652  047637  000000  001176          BIC     @(SP),$TMP1     ;MASK   RMER1
051660  062716  000002                  ADD     #2,(SP)         ;MOVE SP TO NEXT MASK
051664  013737  001376  001200          MOV     RMER2I,$TMP2    ;STORE RMER2 AT TEMP STORAGE
051672  047637  000000  001200          BIC     @(SP),$TMP2     ;MASK RMER2
                                
                                
                                ;CLEAR USER'S ERROR CALL
051700  062716  000006                  ADD     #6,(SP)         ;MOVE SP TO USER'S ERROR CALL
051704  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
051710  162716  000004                  SUB     #4,(SP)         ;LEAVE SP AT NO ERROR RETURN
                                
                                ;SEE IF THERE WERE ANY ERRORS IN RMER1, I.E., $TMP1
051714  005737  001176                  TST     $TMP1           ;ANY ERRORS TO REPORT??
051720  001420                          BEQ     5$              ;NO !!
051722  013737  001176  001142          MOV     $TMP1,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
051730  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
051734  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
051740  112776  000066  000000          MOVB    #66,@(SP)       ;CORRECTABLE DATA CHECK ERROR #
051746  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
051752  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
051754  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
051760  000240                          NOP
051762                          5$:
                                
                                
                                ;SEE IF THERE ARE ANY ERRORS TO REPORT IN RMER2 ($TMP2)
051762  005737  001200                  TST     $TMP2           ;ANY ERRORS IN RMER2?
051766  001420                          BEQ     10$             ;NO!!
                                
051770  013737  001200  001142          MOV     $TMP2,$BDDAT    ;RECEIVED STATUS FOR TYPEOUT
051776  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS FOR TYPEOUT
052002  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
052006  112776  000067  000000          MOVB    #67,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
```

```
     58 052014  16. 6 000002                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     59 052020  004756                      JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     60 052022  162716  000010              SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     61 052026  000240                      NOP
     62 052030                      10$:
     63
     64                             ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
     65 052030  062716  000004              ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     66 052034  105776  000000              TSTB    @(SP)           ;WAS THERE AN ERROR CALLED??
     67 052040  001403                      BEQ     20$             ;NO!.
     68 052042  062716  000004              ADD     #4,(SP)         ;YES - MOVE SP TO ERROR RETURN
     69 052046  000402                      BR      30$
     70 052050  162716  000004      20$:    SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
     71 052054  000207              30$:    RTS     PC              ;RETURN TO USER
     72
```

```
                        .SBTTL  DEVICE SELECT SUBROUTINE

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

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

052056                  DEVSEL:

                        ;CLEAR USER'S ERROR CALL
052056  062716  000004          ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
052062  105076  000000          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE OF CALL
052066  162716  000004          SUB     #4,(SP)         ;MOVE SP BACK

                        ;SAVE USER'S INFORMATION AND SETUP REGISTERS
052072  013746  000004          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
052076  013746  000006          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
052102  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
052104  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052106  012737  052226  000004  MOV     #20$,ERRVEC     ;SETUP FOR BUS TIMEOUT
052114  012737  000300  000006  MOV     #PR6,ERRVEC+2
052122  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
052126  013701  001464          MOV     TSTQUE,R1       ;R1 POINTS TO DEVICE NUMBER

                        ;SELECT DEVICE AND VERIFY THAT DEVICE IS AVAILABLE
052132  111160  000010          MOVB    (R1),RMCS2(R0)  ;WRITE UNIT SELECT BITS
052136  016037  000000  001176  MOV     RMCS1(R0),$TMP1 ;GET "DVA" STATUS
052144  016037  000010  001174  MOV     RMCS2(R0),$TMP0 ;GET "NED" STATUS

052152  032737  010000  001174  BIT     #NED,$TMP0      ;IS DEVICE NONEXISTENT ?
052160  001407                  BEQ     10$             ;NO!!
052162  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
052170  112776  000111  000010  MOVB    #111,@10(SP)    ;WRITE ERROR NUMBER
052176  000422                  BR      30$

052200  032737  004000  001176  10$:    BIT     #DVA,$TMP1      ;IS DEVICE AVAILABLE ?
052206  001021                  BNE     35$             ;YES!!
052210  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
052216  112776  000112  000010  MOVB    #112,@10(SP)    ;WRITE ERROR NUMBER
052224  000407                  BR      30$

                        ;HANDLE BUS TIMEOUT
052226  022626          20$:    CMP     (SP)+,(SP)+     ;ADJUST SP
052230  062766  000004  000010  ADD     #4,10(SP)       ;MOVE SP TO USERS ERROR CALL
052236  112776  000113  000010  MOVB    #113,@10(SP)    ;WRITE BUS TIMEOUT ERROR NUMBER
052244  162766  000002  000010  30$:    SUB     #2,10(SP)       ;ADJUST RETURN TO 'NOP' PRECEDING
                                                        ;THE ERROR CALL

                        ;RESTORE USERS DATA AND RETURN TO ADDRESS ON STACK
052252                  35$:
052252  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
052254  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
052256  012637  000006          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
```

```
        052262  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
     52 052266  000207                          RTS     PC              ;EXIT
     53
```

```
                                .SBTTL  SEEK STATUS CHECK SUBROUTINE

                                ;THIS SUBROUTINE VERIFIES THE RESULTS OF SEEK TESTS USING STATUS
                                ;STORED IN THE GET BUFFER AND TEST PARAMETERS STORED IN THE PUT BUFFER.

                                ;
                                ;THE SUBROUTINE RETURNS TO THE CALLING ROUTINE IF AN ERROR IS DETECTED
                                ;AFTER HAVING LOADED THE APPROPRIATE ERROR NUMBER IN THE "ERROR" TRAP
                                ;OF THE CALLING ROUTINE.  SEEK STATUS IS CHECKED AS FOLLOWS:
                                ;
                                ;CALL:
                                ;(1)     JSR     PC,SEKSTS
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS

052270                          SEKSTS:

                                ;CLEAR USERS' ERROR CALL
052270  000240                          NOP
052272  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
052276  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
052302  162716  000004                  SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
052306  005037  053526                  CLR     300$            ;CLEAR ERROR FLAGS

                                ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN WRITING
                                ;LOCAL REGISTERS, I.E., "PAR"  = 1 AND "DPE" = 0
052312  032737  000010  001350          BIT     #PAR,RMER1I     ;WAS PARITY ERROR DETECTED??
052320  001424                          BEQ     1$              ;NO.!
052322  032737  000010  001376          BIT     #DPE,RMER2I     ;WAS IT DUE TO CONTROL BUS??
052330  001020                          BNE     1$              ;NOT SURE!!

                                ;REPORT REGISTER PARITY ERROR VIA USER'S ERROR CALL
052332  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
052336  013737  001350  001142          MOV     RMER1I,$BDDAI   ;RECEIVED STATUS
052344  062716  000004                  ADD     #4,(SP)         ;MOVE STACK TO USER'S ERROR
052350  112776  000050  000000          MOVB    #50,@(SP)       ;ERROR #50
052356  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052362  004736                          JSR     PC,@(SP)+
052364  162716  000010                  SUB     #10,(SP)        ;RESTORE STACK
052370  000437                          BR      3$              ;IAE SHOULD  BE ZERO

                                ;DETERMINE THE VALUE OF "IAE" STATUS BASED ON TRACK, SECTOR AND CYLINDER
                                ;ALSO, SET "SKI" IF CYLINDER ADDRESS IS TOO LARGE.
052372  012737  002000  001140  1$:     MOV     #IAE,$GDDAT     ;SETUP FOR IAE = 1
052400  052737  040000  053526          BIS     #SKI,300$       ;SETUP FOR SKI = 1
052406  023727  001444  001466          CMP     RMDC0,#822.     ;GREATER THAN LAST CYLINDER ?
052414  101025                          BHI     3$              ;YES - CYLINDER IS INVALID
052416  042737  040000  053526          BIC     #SKI,300$       ;CLEAR SKI ERROR FLAG

052424  123737  001417  001333          CMPB    RMDA0+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
052432  101016                          BHI     3$              ;YES - TRACK IS INVALID

052434  123727  001416  000035          CMPB    RMDA0,#29.      ;SECTOR > 29. ?
052442  101410                          BLOS    2$              ;BR IF NO
```

```
     058 052444  032737  010000  001442          BIT     #FMT16,RMOFO    ;18 BIT FORMAT ?
     059 052452  001406                          BEQ     3$              ;YES - SECTOR IS INVALID FOR 18 BIT MODE
     060 052454  123727  001416  000037          CMPB    RMDAO,#31.      ;SECTOR > 31. ?
     061 052462  101002                          BHI     3$              ;YES - SECTOR IS INVALID
     062
     063 052464  005037  001140          2$:     CLR     $GDDAT          ;"IAE" SHOULD = 0
     064
     065                                 ;COMPARE EXPECTED AND RECIEVED "IAE" STATUS
     066 052470  013737  001350  001142  3$:     MOV     RMER1I,$BDDAT   ;IS IAE OK??
     067 052476  042737  175777  001142          BIC     #^C1AE,$BDDAT   ;SAVE IAE BIT FOR COMPARE
     068 052504  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;CORRECT "IAE" STATUS ?
     069 052512  001004                          BNE     35$             ;BR IF NO
     070 052514  042737  040000  053526          BIC     #SKI,300$       ;CLEAR SKI FLAG
     071 052522  000413                          BR      5$              ;GO CHECK NEXT ERROR
     072 052524                          35$:
     073                                 ;REPORT INCORRECT "IAE" STATUS VIA USER'S ERROR CALL
     074 052524  062716  000004                  ADD     #4,(SP)
     075 052530  112776  000051  000000          MOVB    #51,@(SP)       ;ERROR 51
     076 052536  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     077 052542  004736                          JSR     PC,@(SP)+       ;REPORT INCORRFCT IAE
     078 052544  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP)
     079 052550  000240                          NOP
     080 052552                          5$:
     081
     082                                 ;REPORT ANY IVC ERROR AS
     083                                 ;       IVC ERROR WITH VOLUME VALID ZERO
     084                                 ;       ERRONEOUS IVC ERROR, VOLUME VALID IS SET
     085 052552  032737  010000  001376          BIT     #IVC,RMER2I     ;IVC ERROR??
     086 052560  001427                          BEQ     52$             ;NO!!
     087 052562  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
     088 052566  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     089 052574  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR
     090 052600  112776  000060  000000          MOVB    #60,@(SP)       ;ERROR 60 IF VV - 0
     091 052606  032737  000100  001346          BIT     #VV,RMDSI
     092 052614  001403                          BEQ     51$
     093 052616  112776  000061  000000          MOVB    #61,@(SP)       ;ERROR 61 IF VV  = 1
     094 052624  162716  000002          51$:    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     095 052630  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     096 052632  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
     097 052636  000240                          NOP
     098
     099 052640  013737  001376  001142  52$:    MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     100 052646  042737  137777  001142          BIC     #^CSKI,$BDDAT   ;CLEAR DONT CARES
     101 052654  013737  053526  001140          MOV     300$,$GDDAT     ;GET EXPECTED SKI STATUS
     102 052662  042737  137777  001140          BIC     #^CSKI,$GDDAT   ;CLEAR DONT CARES
     103 052670  001417                          BEQ     53$             ;BRANCH IF 0 EXPECTED
     104
     105                                 ;REPORT ERROR IF SKI IS NOT SET (IAE WAS NOT DETECTED)
     106 052672  032737  040000  001142          BIT     #SKI,$BDDAT     ;WAS SKI DETECTED ??
     107 052700  001032                          BNE     54$             ;YES !!
     108 052702  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
     109 052706  112776  000267  000000          MOVB    #267,@(SP)      ;WRITE ERROR NUMBER
     110 052714  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO ERROR RETURN
     111 052720  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
     112 052722  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR
     113 052726  000443                          BR      6$              ;GO TO NEXT ERROR CHECK
     114 052730                          53$:
```

```
                                        ;REPORT ERROR IF SKI IS SET
052730  032737  040000  001142          BIT     #SKI,$BDDAT     ;IS SKI SET ??
052736  001413                          BEQ     54$             ;NO - SKI IS OK
052740  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
052744  112776  000054  000000          MOVB    #54,@(SP)       ;LOAD ERROR NUMBER
052752  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
052756  004736                          JSR     PC,@(SP)+       ;REPORT SEEK ERROR
052760  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
052764  000240                          NOP

                                        ;REPORT ANY DEVICE CHECK
052766  032737  000200  001376  54$:    BIT     #DVC,RMER2I     ;WAS THERE DVC DURING SEEK.?
052774  001420                          BEQ     6$              ;NO.!
052776  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
053002  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
053010  062716  000004                  ADD     #4,(SP)
053014  112776  000055  000000          MOVB    #55,@(SP)       ;ERROR  #55
053022  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053026  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053030  162716  000010                  SUB     #10,(SP)        ;RESTORE SP
053034  000240                          NOP

                                        ;REPORT ANY "OPI" ERROR AS OPI WITH MOL  = 0, OR  OPI
                                        ;BECAUSE ON CYLINDER LATCH DIDN'T RESET
053036  032737  020000  001350  6$:     BIT     #OPI,RMER1I     ;"OPI" ERROR??
053044  001427                          BEQ     8$              ;NO!!
053046  005037  001140                  CLR     $GDDAT          ;EXPECTED STATUS
053052  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
053060  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
053064  112776  000052  000000          MOVB    #52,@(SP)       ;LOAD ERROR NUMBER
053072  032737  010000  001346          BIT     #MOL,RMDSI      ;WAS MEDIUM ON LINE??
053100  001403                          BEQ     7$              ;NO!!
053102  112776  000053  000000          MOVB    #53,@(SP)       ;YES - CHANGE ERROR NUMBER
053110  162716  000002          7$:     SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053114  004736                          JSR     PC,@(SP)+       ;REPORT "OPI" ERROR
053116  162716  000010                  SUB     #10,(SP)        ;RESTORE (SP)
053122  000240                          NOP

                                        ;SEE IF "PIP" = 0, AND "ATA", "MOL" AND "VV" = 1
053124  013746  001346          8$:     MOV     RMDSI,-(SP)
053130  042716  047677                  BIC     #^C<ATA!PIP!MOL!VV>,(SP)
053134  022726  110100                  CMP     #ATA!MOL!VV,(SP)+
053140  001002                          BNE     9$              ;ERROR IN RMDS
053142  000137  053476                  JMP     14$             ;RMDS IS OK

                                        ;REPORT ERROR IF MOL = 0 AND OPI = 0
053146  032737  010000  001346  9$:     BIT     #MOL,RMDSI      ;IS MOL RESET??
053154  001030                          BNE     10$             ;NO - MOL IS SET
053156  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET
053164  001024                          BNE     10$             ;YES - DONT REPORT ERROR
053166  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053174  052737  010000  001140          BIS     #MOL,$GDDAT
053202  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053210  062716  000004                  ADD     #4,(SP)
053214  112776  000062  000000          MOVB    #62,@(SP)
053222  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
```

```
053226  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053230  162716  000010                  SUB     #10,(SP)
053234  000240                          NOP

                                ;REPORT AN ERROR IF "PIP" IS STIL SET AND SKI NOT SET
053236  032737  020000  001346  10$:    BIT     #PIP,RMDSI      ;IS "PIP" STILL SET??
053244  001430                          BEQ     11$             ;NO!!
053246  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS "SKI"SET??
053254  001024                          BNE     11$             ;YES-DONT REPORT PIP
053256  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053264  042737  020000  001142          BIC     #PIP,$BDDAT
053272  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053300  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
053304  112776  000056  000000          MOVB    #56,@(SP)       ;LOAD ERROR NUMBER
053312  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053316  004736                          JSR     PC,@(SP)+       ;REPORT "PIP" SET AFTER SEEK
053320  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP)
053324  000240                          NOP

                                ;REPORT AN ERROR IF "ATA" IS NOT SET
053326  032737  100000  001346  11$:    BIT     #ATA,RMDSI      ;WAS "ATA" SET ??
053334  001024                          BNE     13$             ;YES!!
053336  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053344  052737  110600  001140          BIS     #ATA!MOL!DPR.DRY,$GDDAT
053352  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053360  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR
053364  112776  000057  000000          MOVB    #57,@(SP)       ;LOAD ERROR NUMBER
053372  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053376  004736                          JSR     PC,@(SP)+       ;REPORT ATTENTION NOT SET DURING
                                                                ;SEEK TEST
053400  162716  000010                  SUB     #10,(SP)                ;RESTORE (SP)
053404  000240                          NOP

                                ;REPORT ERROR IF VOLUME VALID IS RESET AND IVC IS ZERO
053406  032737  000100  001346  13$:    BIT     #VV,RMDSI       ;IS VV = 0 ??
053414  001030                          BNE     14$             ;NO!!
053416  032737  010000  001376          BIT     #IVC,RMER2I     ;IS IVC ALSO 0 ??
053424  001024                          BNE     14$             ;NO - IVC IS SET
053426  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
053434  052737  000100  001140          BIS     #VV,$GDDAT
053442  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
053450  062716  000004                  ADD     #4,(SP)
053454  112776  000064  000000          MOVB    #64,@(SP)       ;ERROR #64
053462  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053466  004736                          JSR     PC,@(SP)+
053470  162716  000010                  SUB     #10,(SP)
053474  000240                          NOP
053476                          14$:

                                ;MODIFY THE RETURN ADDRESS IF AN ERROR WAS DETECTED
053476  000240                          NOP
053500  062716  000004                  ADD     #4,(SP)         ;MOVE (SP) TO ERROR CALL
053504  105776  000000                  TSTB    @(SP)           ;WAS ERROR CALLED??
053510  001403                          BEQ     15$             ;NO!.
053512  062716  000004                  ADD     #4,(SP)         ;MOVE TO ERROR RETURN
053516  000402                          BR      16$

```

```
    229 053520  162716  000004          15$:    SUB     #4,(SP)         ;MOVE (SP) TO NO ERROR RETURN
    230 053524  000207                  16$:    RTS     PC              ;RETURN
    231
    232 053526  000000                  300$:   .WORD   0               ;ERROR FLAGS
    233
```

```
                                .SBTTL  CONTROLLER CLEAR SUBROUTINE

                                ;THIS SUBROUTINE CLEARS THE MASSBUS CONTROLLER, MASSBUS ADAPTERS,
                                ;AND DRIVES, THEN SELECTS THE DRIVE.
                                ;
                                ;CALL:  JSR     PC,CNTCLR
                                ;       BR      ???             RETURN HERE IF NO ERROR FOUND
                                ;       NOP                     RETURN HERE IF ANY ERROR FOUND
                                ;       ERROR                   SUB DEFINES ERROR NUMBER
                                ;       ???

                                CNTCLR:
053530  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
053532  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
053534  013746  000004                    MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
053540  013746  000006                    MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
053544  012737  053604  000004            MOV     #10$,ERRVEC     ;SETUP FOR BUS TIMEOUT
053552  012737  000300  000006            MOV     #PR6,ERRVEC+2
053560  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS
053564  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR MASSBUS
053572  013701  001464                    MOV     TSTQUE,R1       ;GET DEVICE UNDER TEST
053576  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT DEVICE
053602  000412                            BR      20$

053604  022626                    10$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
053606  062766  000004  000010            ADD     #4,10(SP)       ;MOVE SP TO USER'S ERROR CALL
053614  112776  000007  000010            MOVB    #7,@10(SP)      ;WRITE THE ERROR NUMBER
053622  162766  000002  000010            SUB     #2,10(SP)       ;ADJUST SP TO RETURN TO ERROR
053630                              20$:
053630  012637  000006                    MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
053634  012637  000004                    MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
053640  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
053642  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
053644  000207                            RTS     PC

```

```
                                .SBTTL  STATUS CHECK SUBROUTINES
                                ;;******************************************************************
                                .SBTTL  CONTROLLER CLEAR STATUS CHECK SUBROUTINE

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

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

053646                          CLRSTS:

                                ;CLEAR USER'S ERROR CALL
053646  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR
053652  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
053656  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR
                                ;REPORT ERROR IF RMCS1 NOT INITIALIZED
053662  013737  001334  001142 4$:      MOV     RMCS1I,$BDDAT   ;VERIFY RMCS1
053670  042737  100000  001142          BIC     #SC,$BDDAT      ;IGNORE SPECIAL CONDITION
053676  012737  004200  001140          MOV     #DVA!RDY,$GDDAT ;EXPECT DVA & RDY
053704  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED, RECEIVED
053712  001413                          BEQ     5$              ;BRANCH IF EQUAL
053714  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053720  112776  000126  000000          MOVB    #126,@(SP)      ;WITE ERROR NUMBER IN CALL
053726  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053732  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053734  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
053740  000240                          NOP
                                ;REPORT ERROR IF RMBA NOT RESET
053742  005037  001140          5$:     CLR     $GDDAT          ;VERIFY RMBA IS ZERO
053746  013737  001340  001142          MOV     RMBAI,$BDDAT
053754  001413                          BEQ     7$              ;BRANCH IF ZERO
053756  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
053762  112776  000127  000000          MOVB    #127,@(SP)      ;WITE ERROR NUMBER IN CALL
053770  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
053774  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
053776  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
054002  000240                          NOP
                                ;REPORT ERROR IF RMCS2 NOT INITIALIZED
054004  013737  001344  001142 7$:      MOV     RMCS2I,$BDDAT   ;VERIFY RMCS2
054012  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
054014  005046                          CLR     -(SP)           ;EXPECT IR & UNIT NUMBER
054016  013701  001464                  MOV     TSTQUE,R1       ;R1 = ADDRESS OF TEST QUE
054022  111116                          MOVB    (R1),(SP)
```

```
 58 054024  052716  000100                    BIS     #IR,(SP)
 59 054030  012637  001140                    MOV     (SP)+,$GDDAT
 60 054034  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 61 054036  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 62 054044  001413                            BEQ     9$              ;BRANCH IF EQUAL
 63 054046  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 64 054052  112776  000130  000000            MOVB    #130,@(SP)      ;WITE ERROR NUMBER IN CALL
 65 054060  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 66 054064  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 67 054066  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 68 054072  000240                            NOP
 69                                 ;REPORT ERROR IF RMER1 NOT RESET-IGNORE UNS
 70 054074  005037  001140          9$:     CLR     $GDDAT          ;VERIFY RMER1
 71 054100  013737  001350  001142            MOV     RMER11,$BDDAT
 72 054106  042737  040000  001142            BIC     #UNS,$BDDAT     ;IGNORE UNSAFE
 73 054114  001413                            BEQ     13$             ;BRANCH IF ZERO
 74 054116  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 75 054122  112776  000131  000000            MOVB    #131,@(SP)      ;WITE ERROR NUMBER IN CALL
 76 054130  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 77 054134  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 78 054136  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 79 054142  000240                            NOP
 80                                 ;REPORT ERROR IF RMMR1 NOT INITIALIZED-IGNORE WC,LS,LST
 81 054144  013737  001360  001142  13$:     MOV     RMMR11,$BDDAT   ;VERIFY RMMR
 82 054152  042737  000046  001142            BIC     #WC!LS!LST,$BDDAT       ;IGNORE WORD CLOCK, SCT, TRK
 83 054160  012737  000010  001140            MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA BIT
 84 054166  023737  001140  001142            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
 85 054174  001413                            BEQ     17$             ;BRANCH IF 0
 86 054176  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 87 054202  112776  000133  000000            MOVB    #133,@(SP)      ;WITE ERROR NUMBER IN CALL
 88 054210  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 89 054214  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
 90 054216  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
 91 054222  000240                            NOP
 92                                 ;REPORT AN ERROR IF RMEC2 IS NOT RESET
 93 054224  005037  001140          17$:    CLR     $GDDAT          ;EXPECT ZEROS
 94 054230  013737  001402  001142            MOV     RMEC21,$BDDAT   ;VERIFY RMEC2 = 0
 95 054236  001413                            BEQ     19$
 96 054240  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 97 054244  112776  000135  000000            MOVB    #135,@(SP)      ;WITE ERROR NUMBER IN CALL
 98 054252  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 99 054256  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
100 054260  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
101 054264  000240                            NOP
102                                 ;REPORT ERROR IF RMMR2 NOT INITIALIZED-IGNORE RQA,RQB
103 054266  013737  001374  001142  19$:     MOV     RMMR21,$BDDAT   ;VERIFY RMMR2
104 054274  042737  140000  001142            BIC     #RQA!RQB,$BDDAT
105 054302  012737  011777  001140            MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
106 054310  023737  001140  001142            CMP     $GDDAT,$BDDAT
107 054316  001413                            BEQ     21$
108 054320  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
109 054324  112776  000136  000000            MOVB    #136,@(SP)      ;WITE ERROR NUMBER IN CALL
110 054332  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
111 054336  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
112 054340  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
113 054344  000240                            NOP
114                                 ;REPORT ERROR IF RMER2 NOT RESET-IGNORE SKI,DVC
```

```
115 054346  005037  001140          21$:    CLR     $GDDAT          ;EXPECT ALL ZEROS
116 054352  013737  001376  001142          MOV     RMER2I,$BDDAT   ;VERIFY RMER2
117 054360  042737  040200  001142          BIC     #SKI!DVC,$BDDAT ;IGNORE DEVICE ERRORS
118 054366  001413                          BEQ     215$            ;BRANCH IF OTHER BITS 0
119 054370  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
120 054374  112776  000174  000000          MOVB    #174,@(SP)      ;WITE ERROR NUMBER IN CALL
121 054402  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
122 054406  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
123 054410  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
124 054414  000240                          NOP
125                                 ;REPORT ERROR IF RMDS NOT INITIALIZED
126 054416  013737  001346  001142  215$:   MOV     RMDSI,$BDDAT    ;TEST DRIVE STATUS REGISTER
127 054424  042737  177177  001142          BIC     #^C<DRY!DPR>,$BDDAT
128 054432  012737  000600  001140          MOV     #DPR!DRY,$GDDAT ;EXPECTED DRIVE STATUS
129 054440  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
130 054446  001413                          BEQ     22$             ;BRANCH IF EQUAL
131 054450  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
132 054454  112776  000134  000000          MOVB    #134,@(SP)      ;WRITE ERROR NUMBER
133 054462  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROP
134 054466  004736                          JSR     PC,@(SP)+       ;REPORT ERROR TO USER
135 054470  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
136 054474  000240                          NOP
137 054476  062716  000004          22$:    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
138 054502  105776  000000                  TSTB    @(SP)           ;WAS AN ERROE DETECTED??
139 054506  001403                          BEQ     23$             ;NO!!
140 054510  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
141 054514  000402                          BR      24$
142 054516  162716  000004          23$:    SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
143 054522  000240                  24$:    NOP
144 054524  000207                          RTS     PC
145
```

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

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

054526                                  ACKSTS:

                                        ;CLEAR USER'S ERROR CALL
054526  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054532  105076  000000                          CLRB    @(SP)           ;CLEAR LOW ORDER BYTE
054536  162716  000004                          SUB     #4,(SP)         ;MOVE SP BACK

                                        ;REPORT AN ERROR  IF "VV" IS 0
054542  032737  000100  001346                  BIT     #VV,RMDSI       ;IS VOLUME VALID SET??
054550  001024                                  BNE     1$              ;YES!!
054552  013737  001346  001140                  MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
054560  052737  000100  001140                  BIS     #VV,$GDDAT
054566  013737  001346  001142                  MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
054574  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054600  112776  000155  000000                  MOVB    #155,@(SP)      ;WRITE NUMBER IN ERROR CALL
054606  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054612  004736                                  JSR     PC,@(SP)+       ;REPORT THE ERROR
054614  162716  000010                          SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
054620  000240                                  NOP
054622                                  1$:

                                        ;REPORT AN ERROR IF "MOL" IS 0
054622  032737  010000  001346                  BIT     #MOL,RMDSI      ;IS MOL SET??
054630  001024                                  BNE     2$              ;YES!!
054632  013737  001346  001140                  MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
054640  052737  010000  001140                  BIS     #MOL,$GDDAT
054646  013737  001346  001142                  MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
054654  062716  000004                          ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
054660  112776  000041  000000                  MOVB    #41,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
054666  162716  000002                          SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
054672  004736                                  JSR     PC,@(SP)+       ;REPORT TH ERROR
054674  162716  000010                          SUB     #10,(SP)        ;MOVE SP TO BRANCH
054700  000240                                  NOP
054702                                  2$:

                                        ;SEE IF "UNS","OPI","RMR","ILR", OR "ILF" IS SET
054702  032737  060007  001350                  BIT     #UNS!OPI!RMR!ILR!ILF,RMER1I
054710  001570                                  BEQ     7$

                                        ;REPORT AN ERROR IF "UNS" IS SET
054712  032737  040000  001350                  BIT     #UNS,RMER1I     ;WAS UNS SET??
054720  001424                                  BEQ     3$              ;NO!
054722  013737  001350  001142                  MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
```

```
 58 054730  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 59 054736  042737  040000  001140          BIC     #UNS,$GDDAT
 60 054744  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 61 054750  112776  000042  000000          MOVB    #42,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
 62 054756  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 63 054762  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 64 054764  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 65 054770  000240                          NOP
 66 054772                          3$:
 67
 68                                 ;REPORT ANY OPI ERROR
 69 054772  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET ??
 70 055000  001424                          BEQ     4$              ;NO!!
 71 055002  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 72 055010  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 73 055016  042737  020000  001140          BIC     #OPI,$GDDAT
 74 055024  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 75 055030  112776  000043  000000          MOVB    #43,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
 76 055036  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 77 055042  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 78 055044  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 79 055050  000240                          NOP
 80 055052                          4$:
 81
 82                                 ;REPORT ANY RMR ERROR
 83 055052  032737  000004  001350          BIT     #RMR,RMER1I     ;WAS RMR SET??
 84 055060  001424                          BEQ     5$              ;NO!!
 85 055062  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
 86 055070  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
 87 055076  042737  000004  001140          BIC     #RMR,$GDDAT
 88 055104  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 89 055110  112776  000044  000000          MOVB    #44,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
 90 055116  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 91 055122  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 92 055124  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 93 055130  000240                          NOP
 94 055132                          5$:
 95
 96                                 ;REPORT ANY     ILR ERROR
 97 055132  032737  000002  001350          BIT     #ILR,RMER1I     ;WAS ILR SET??
 98 055140  001424                          BEQ     6$              ;NO!!
 99 055142  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
100 055150  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
101 055156  042737  000002  001140          BIC     #ILR,$GDDAT
102 055164  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
103 055170  112776  000045  000000          MOVB    #45,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
104 055176  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
105 055202  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
106 055204  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
107 055210  000240                          NOP
108 055212                          6$:
109
110                                 ;REPORT ANY     ILF ERROR
111 055212  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS ILF SET??
112 055220  001424                          BEQ     7$              ;NO!!
113 055222  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
114 055230  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
```

```
    115 055236  042737  000001  001140          BIC     #ILF,$GDDAT
    116 055244  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    117 055250  112776  000046  000000          MOVB    #46,@(SP)       ;WRITE NUMBER OF ERROR IN CALL
    118 055256  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    119 055262  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
    120 055264  162716  000010                  SUB     #10.,(SP)       ;MOVE SP TO NO ERROR RETURN
    121 055270  000240                          NOP
    122 055272                          7$:
    123
    124                                 ;AUGMENT RETURN ADDRESS IF ERROR WAS FOUND
    125 055272  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
    126 055276  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
    127 055302  001403                          BEQ     8$              ;NO!!
    128 055304  062716  000004                  ADD     #4,(SP)         ;YES - MOVE TO ERROR RETURN
    129 055310  000402                          BR      9$
    130 055312  162716  000004          8$:     SUB     #4,(SP)         ;MOVE SP TO NO ERROR RETURN
    131 055316  000240                  9$:     NOP
    132 055320  000207                          RTS     PC
    133
```

```
                                .SBTTL  RECALIBRATE STATUS CHECK SUBROUTINE
                                
                                ;THIS SUBROUTINE CHECKS THE RESULTS OF A RECALIBRATE OPERATION
                                ;USING THE STATUS STORED IN THE GET BUFFER.
                                
                                ;CALL:
                                ;
                                ;(1)    JSR     PC,RCLSTS       ;CALL SUBROUTINE
                                ;       BR      ???             RETURN HERE IF NO ERROR
                                ;       NOP                     RETURN HERE TO REPORT AN ERROR
                                ;       ERROR                   ERROR NUMBER DEFINED BY SUB
                                ;       JSR     PC,@(SP)+       GO BACK TO SUB FOR MORE ERROR CHECKS
                                ;       ???                     RETURN HERE IF NO MORE ERRORS
                                
055322                          RCLSTS:
                                
                                ;CLEAR USER'S ERROR NUMBER
055322  062716  000004                  ADD     #4,(SP)
055326  105076  000000                  CLRB    @(SP)           ;CLEAR USER'S ERROR CALL
055332  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO BRANCH
                                
                                
                                ;SEF IF "PAR" OR "ILF" OR "OPI" OR "IAE" IS SET
055336  032737  022011  001350          BIT     #OPI!PAR.ILF.IAE,RMER1I
055344  001553                  BEQ     4$              ;NONE ARE SET - GO TO NEXT CHECK
                                
                                ;REPORT ANY MASSBUS CONTROL BUS PARITY ERROR, I.E.,
                                ;"PAR" = 1 AND "DPE" = 0
055346  032737  000010  001350          BIT     #PAR,RMER1I     ;WAS "PAR" SET??
055354  001430                  BEQ     1$              ;NO!!
055356  032737  000010  001376          BIT     #DPE,RMER2I     ;WAS "DPE" SET??
055364  001024                  BNE     1$              ;YES - NOT A REGISTER ERROR
055366  013737  001350  001140          MOV RMER1I,$GDDAT       ;EXPECTED STATUS
055374  042737  000010  001140          BIC     #PAR,$GDDAT
055402  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055410  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
055414  112776  000050  000000          MOVB    #50,@(SP)       ;WRITE ERROR NUMBER IN CALL
055422  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055426  004736                  JSR     PC,@(SP)+       ;GO REPORT ERROR
055430  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
055434  000240                  NOP
055436                          1$:
                                
                                ;REPORT ANY "ILF" ERROR
055436  032737  000001  001350          BIT     #ILF,RMER1I     ;WAS "ILF" SET??
055444  001424                  BEQ     2$              ;NO!!
055446  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055454  042737  000001  001140          BIC     #ILF,$GDDAT
055462  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055470  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055474  112776  000071  000000          MOVB    #71,@(SP)       ;WRITE ERROR NUMBER IN CALL
055502  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055506  004736                  JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055510  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
055514  000240                  NOP
055516                          2$:
```

```
                                        ;REPORT ANY "OPI" ERROR AS
                                        ;       . OPI DUE TO "MOL" = 0
                                        ;       . OPI BECAUSE ON CYLINDER LATCH DIDN'T RESET
055516  032737  020000  001350            BIT     #OPI,RMER1I     ;WAS OPI SET??
055524  001433                            BEQ     31$             ;NO!!
055526  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055534  042737  020000  001140            BIC     #OPI,$GDDAT
055542  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055550  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055554  112776  000072  000000            MOVB    #72,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
055562  032737  010000  001346            BIT     #MOL,RMDSI      ;WAS "MOL" = 0??
055570  001403                            BEQ     3$              ;YES!!
055572  112776  000073  000000            MOVB    #73,@(SP)       ;NO - CHANGE ERROR NUMBER
055600  162716  000002          3$:       SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055604  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055606  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
055612  000240                            NOP
055614                          31$:

                                        ;REPORT AN ERROR IF "IAE" IS SET
055614  032737  002000  001350            BIT     #IAE,RMER1I     ;IS "IAE" SET??
055622  001424                            BEQ     4$              ;NO!!
055624  013737  001350  001140            MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
055632  042737  002000  001140            BIC     #IAE,$GDDAT
055640  013737  001350  001142            MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
055646  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
055652  112776  000070  000000            MOVB    #70,@(SP)       ;WRITE ERROR NUMBER IN USER'S CALL
055660  162716  000002                    SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055664  004736                            JSR     PC,@(SP)+       ;REPORT ERROR
055666  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
055672  000240                            NOP
055674                          4$:

                                        ;SEE IF "SKI" OR "IVC" OR "DVC" IS SET
055674  032737  050200  001376            BIT     #SKI!IVC!DVC,RMER2I
055702  001517                            BEQ     8$              ;NONE OF THE BITS ARE SET


                                        ;REPORT ANY "IVC" ERROR AS
                                        ;       .IVC WITH VV = 0
                                        ;       .ERRONEOUS IVC ERROR
055704  032737  010000  001376            BIT     #IVC,RMER2I     ;WAS IVC SET??
055712  001433                            BEQ     6$              ;NO!!
055714  013737  001376  001140            MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
055722  042737  010000  001140            BIC     #IVC,$GDDAT
055730  013737  001376  001142            MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
055736  062716  000004                    ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
055742  112776  000074  000000            MOVB    #74,@(SP)       ;WRITE ERROR NUMBER IN CALL
055750  032737  000100  001346            BIT     #VV,RMDSI       ;WAS VV = 0??
055756  001403                            BEQ     5$              ;YES!!
055760  112776  000075  000000            MOVB    #75,@(SP)       ;NO - CHANGE ERROR NUMBER
055766  162716  000002          5$:       SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
055772  004736                            JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
055774  162716  000010                    SUB     #10,(SP)        ;MOVE SP BACK TO BRANCH
056000  000240                            NOP
056002                          6$:

```

```
115                                     ;REPORT ANY "SKI" ERROR
116 056002  032737  040000  001376              BIT     #SKI,RMER2I     ;WAS SKI SET??
117 056010  001424                              BEQ     7$              ;NO!!
118 056012  013737  001376  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
119 056020  042737  040000  001140              BIC     #SKI,$GDDAT
120 056026  013737  001376  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
121 056034  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
122 056040  112776  000076  000000              MOVB    #76,@(SP)       ;WRITE ERROR NUMBER
123 056046  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
124 056052  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
125 056054  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO BRANCH
126 056060  000240                              NOP
127 056062                              7$:
128.
129                                     ;REPORT ANY "DVC" ERROR
130 056062  032737  000200  001376              BIT     #DVC,RMER2I     ;WAS "DVC" SET??
131 056070  001424                              BEQ     8$              ;NO!!
132 056072  013737  001376  001140              MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
133 056100  042737  000200  001140              BIC     #DVC,$GDDAT
134 056106  013737  001376  001142              MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
135 056114  062716  000004                      ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
136 056120  112776  000077  000000              MOVB    #77,@(SP)       ;WRITE ERROR NUMBER
137 056126  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
138 056132  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
139 056134  162716  000010                      SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
140 056140  000240                              NOP
141 056142                              8$:
142
143                                     ;SEE IF "PIP" AND "OM" ARE 0, AND "ATA","MOL" AND "VV" ARE 1
144 056142  013746  001346                      MOV     RMDSI,-(SP)     ;PUT RMDS ON STACK
145 056146  042716  047676                      BIC     #^C<PIP!MOL!VV!OM!ATA>,(SP)
146 056152  022726  110100                      CMP     #ATA!MOL!VV,(SP)+
147 056156  001002                              BNE     85$
148 056160  000137  056574                      JMP     13$
149 056164                              85$:
150
151                                     ;REPORT AN ERROR IF MOL = 0 AND OPI = 0, I.E., MEDIUM WENT OFF
152                                     ;LINE AFTER RECALIBRATE WAS INITIATED
153 056164  032737  010000  001346              BIT     #MOL,RMDSI      ;DID MOL DROP??
154 056172  001030                              BNE     9$              ;NO!!
155 056174  032737  020000  001350              BIT     #OPI,RMER1I     ;WAS OPI ERROR REPORTED??
156 056202  001024                              BNE     9$              ;YES - DON'T REPORT MOL = 0
157 056204  013737  001346  001140              MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
158 056212  052737  010000  001140              BIS     #MOL,$GDDAT
159 056220  013737  001346  001142              MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
160 056226  062716  000004                      ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
161 056232  112776  000100  000000              MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
162 056240  162716  000002                      SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
163 056244  004736                              JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
164 056246  162716  000010                      SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
165 056252  000240                              NOP
166 056254                              9$:
167
168                                     ;REPORT AN ERROR IF "VV" = 0 AND "IVC" = 0
169 056254  032737  000100  001346              BIT     #VV,RMDSI       ;DID "VV" DROP??
170 056262  001030                              BNE     10$             ;NO!!
171 056264  032737  010000  001376              BIT     #IVC,RMER2I     ;WAS THERE A IVC ERROR??
```

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

                                        ;REPORT AN ERROR IF ATA IS NOT SET
                                                BIT     #ATA,RMDSI      ;WAS ATA SET DURING RECALIBRATE??
                                                BNE     11$             ;YES!!
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIS     #ATA,$GDDAT
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #102,@(SP)      ;WRITE ERROR NUMBER IN CALL
                                                SUB     #2,(SP)
                                                JSR     PC,@(SP)+
                                                SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
                                                NOP

                                        11$:

                                        ;REPORT AN ERROR IF "OM" IS NOT ZERO BECAUSE RECALIBRATE SHOULD
                                        ;ALWAYS CLEAR OFFSET MODE
                                                BIT     #OM,RMDSI       ;WAS "OM" RESET??
                                                BEQ     12$             ;YES!!
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIC     #OM,$GDDAT
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #103,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
                                                SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
                                                NOP
                                        12$:

                                        ;REPORT AN ERROR IF "PIP" IS STIL ON, I.E., DRIVE NOT ON
                                        ;CYLINDER
                                                BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
                                                BEQ     13$             ;NO!!
                                                BIT     #SKI,RMER2I     ;WAS "SKI" DETECTED??
                                                BNE     13$             ;YES-DONT REPORT "PIP"
                                                MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
                                                BIC     #PIP,$GDDAT
                                                MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
                                                ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
                                                MOVB    #104,@(SP)      ;WRITE ERROR NUMBER
                                                SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
                                                JSR     PC,@(SP)+
                                                SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
                                                NOP
```

```
                                        13$:

                                        ;SEE IF ''ILR'' OR ''RMR'' OR ''UNS'' IS SET
056574  032737  040006  001350                BIT     #ILR.RMR!UNS,RMER1I
056602  001514                                BEQ     16$

                                        ;REPORT AN ERROR IF ''ILR'' IS SET
056604  032737  000002  001350                BIT     #ILR,RMER1I     ;WAS ILR SET DURING RECALIBRATE??
056612  001424                                BEQ     14$             ;NO!.
056614  013737  001350  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056622  042737  000002  001140                BIC     #ILR,$GDDAT
056630  013737  001350  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056636  062716  000004                        ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
056642  112776  000105  000000                MOVB    #105,@(SP)      ;WRITE ERROR NUMBER IN CALL
056650  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056654  004736                                JSR     PC,@(SP)+
056656  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
056662  000240                                NOP
056664                                  14$:

                                        ;REPORT AN ERROR IF ''RMR'' IS SET
056664  032737  000004  001350                BIT     #RMR,RMER1I     ;WAS RMR SET??
056672  001424                                BEQ     15$             ;NO!!
056674  013737  001350  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056702  042737  000004  001140                BIC     #RMR,$GDDAT
056710  013737  001350  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
056716  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
056722  112776  000106  000000                MOVB    #106,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
056730  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
056734  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
056736  162716  000010                        SUB     #10,(SP)        ;MOVE SP TO USER'S BRANCH
056742  000240                                NOP
056744                                  15$:

                                        ;REPORT AN ERROR IF ''UNS'' IS SET AND ''DVC'' IS 0
056744  032737  040000  001350                BIT     #UNS,RMER1I     ;WAS UNSAFE ON??
056752  001430                                BEQ     16$             ;NO!!
056754  032737  000200  001376                BIT     #DVC,RMER2I     ;WAS THERE A DEVICE CHECK??
056762  001024                                BNE     16$             ;YES - DON'T REPORT UNSAFE
056764  013737  001350  001140                MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
056772  042737  040000  001140                BIC     #UNS,$GDDAT
057000  013737  001350  001142                MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
057006  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057012  112776  000107  000000                MOVB    #107,@(SP)      ;WRITE ERROR NUMBER
057020  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
057024  004736                                JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
057026  162716  000010                        SUB     #10,(SP)        ;MOVE SP BACK TO USER'S BRANCH
057032  000240                                NOP
057034                                  16$:

                                        ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
057034  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057040  105776  000000                        TSTB    @(SP)           ;WAS AN ERROR REPORTED??
057044  001403                                BEQ     17$             ;NO!.
057046  062716  000004                        ADD     #4,(SP)         ;YES - AUGMENT SP RETURN
057052  000402                                BR      18$
057054  162716  000004                17$:    SUB     #4,(SP) ;NO ERROR - RETURN SP TO BRANCH
```

```
286 057060  000240                18$:    NOP
287 057062  000207                        RTS     PC              ;STATUS CECK IS COMPLETE
288
```

```
                                        .SBTTL  DRIVE CLEAR STATUS CHECK SUBROUTINE

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

 057064                                 DRVSTS:

                                        ;CLEAR USER'S ERROR CALL
 057064  062716  000004                         ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 057070  105076  000000                         CLRB    @(SP)           ;CLEAR ERROR CALL
 057074  162716  000004                         SUB     #4,(SP)         ;MOVE SP TO USER'S BRANCH
                                        ;REPORT ERROR IF RMCS1 NOT INITIALIZED
 057100  013737  001334  001142  4$:    MOV     RMCS1I,$BDDAT   ;CHECK RMCS1
 057106  042737  173700  001142         BIC     #^C<DVA!FNCMSK>,$BDDAT  ;CLEAR DONT CARES
 057114  012737  004010  001140         MOV     #DVA!DRVCLR,$GDDAT      ;EXPECT DVA
 057122  023737  001140  001142         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 057130  001443                         BEQ     6$              ;BRANCH IF EQUAL
 057132  062716  000004                 ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 057136  112776  000141  000000         MOVB    #141,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 057144  162716  000002                 SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 057150  004736                         JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 057152  162716  000010                 SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 057156  000240                         NOP
                                        ;REPORT ERROR IF RMDS NOT INITIALIZED
 057160  013737  001346  001142  5$:    MOV     RMDSI,$BDDAT    ;CHECK RMDS
 057166  042737  021101  001142         BIC     #PGM!OM!VV!PIP,$BDDAT   ;CLEAR DONT CARES
 057174  012737  010600  001140         MOV     #MOL!DPR!DRY,$GDDAT     ;EXPECT DRY & DPR
 057202  023737  001140  001142         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 057210  001413                         BEQ     6$              ;BRANCH IF EQUAL
 057212  062716  000004                 ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 057216  112776  000142  000000         MOVB    #142,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 057224  162716  000002                 SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 057230  004736                         JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 057232  162716  000010                 SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 057236  000240                         NOP
                                        ;REPORT ERROR IF RMER1 NOT INITIALIZED
 057240  005037  001140         6$:    CLR     $GDDAT          ;EXPECT 0'S
 057244  013737  001350  001142         MOV     RMER1I,$BDDAT   ;CHECK RMER1
 057252  001413                         BEQ     8$              ;BRANCH IF EQUAL
 057254  062716  000004                 ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 057260  112776  000143  000000         MOVB    #143,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 057266  162716  000002                 SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 057272  004736                         JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 057274  162716  000010                 SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 057300  000240                         NOP
                                        ;REPORT ERROR IF ATA NOT INITIALIZED
 057302  013737  001352  001142  8$:    MOV     RMAS1,$BDDAT    ;CHECK ATTENTION BIT
 057310  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 057312  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 057314  013701  001464                 MOV     TSTQUE,R1
 057320  116102  000001                 MOVB    1(R1),R2
 057324  042702  177400                 BIC     #^CATNMSK,R2
 057330  005102                         COM     R2
 057332  040237  001142                 BIC     R2,$BDDAT
```

```
 58 057336  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 59 057340  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 60 057342  005737  001142                  TST     $BDDAT          ;IS ATTENTION CLEARED??
 61 057346  001413                          BEQ     9$              ;BRANCH IF ATTENTION CLEARED
 62 057350  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 63 057354  112776  000144  000000          MOVB    #144,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 64 057362  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 65 057366  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 66 057370  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 67 057374  000240                          NOP
 68                                 ;REPORT ERROR IF RMMR1 NOT INITIALIZED
 69 057376  013737  001360  001142  9$:     MOV     RMMR1I,$BDDAT   ;CHECK RMMR
 70 057404  042737  000046  001142          BIC     #WC!LS!LST,$BDDAT       ;CLEAR DONT CARES
 71 057412  012737  000010  001140          MOV     #MWD,$GDDAT     ;EXPECT WRITE DATA ON
 72 057420  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED AND RECEIVED
 73 057426  001413                          BEQ     11$             ;BRANCH IF ZERO
 74 057430  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 75 057434  112776  000145  000000          MOVB    #145,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 76 057442  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 77 057446  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 78 057450  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 79 057454  000240                          NOP
 80                                 ;REPORT ERROR IF RMMR2 NOT INITIALIZED
 81 057456  013737  001374  001142  11$:     MOV     RMMR2I,$BDDAT   ;CHECK RMMR2
 82 057464  042737  140000  001142          BIC     #RQA!RQB,$BDDAT ;CLEAR REQA, REQB
 83 057472  012737  011777  001140          MOV     #TST!1777,$GDDAT        ;EXPECT TEST BIT ON
 84 057500  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED & RECEIVED
 85 057506  001413                          BEQ     15$             ;BRANCH IF EQUAL
 86 057510  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 87 057514  112776  000146  000000          MOVB    #146,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 88 057522  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 89 057526  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
 90 057530  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
 91 057534  000240                          NOP
 92 057536  005037  001140          15$:    CLR     $GDDAT          ;EXPECT ZEROS
 93                                 ;REPORT ERROR IF RMEC2 NOT RESET
 94 057542  013737  001402  001142          MOV     RMEC2I,$BDDAT   ;CHECK RMEC2
 95 057550  001413                          BEQ     17$             ;BRANCH IF 0
 96 057552  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
 97 057556  112776  000150  000000          MOVB    #150,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
 98 057564  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
 99 057570  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
100 057572  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
101 057576  000240                          NOP
102                                 ;REPORT ERROR IF RMER2 NOT RESET
103 057600  013737  001376  001142  17$:     MOV     RMER2I,$BDDAT   ;CHECK RMER2
104 057606  001413                          BEQ     18$             ;BRANCH IF NO ERROR
105 057610  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO ERROR CALL
106 057614  112776  000147  000000          MOVB    #147,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
107 057622  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
108 057626  004736                          JSR     PC,@(SP)+       ;REPORT THE ERROR VIA USER
109 057630  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
110 057634  000240                          NOP
111 057636                         18$:
112
113 057636                         19$:
114
```

```
115
116 057636  062716  000004          ;AUGMENT RETURN ADDRESS IF ANY ERROR WAS FOUND
                            ADD     #4,(SP)          ;MOVE SP TO ERROR CALL
117 057642  105776  000000          TSTB    @(SP)            ;WAS AN ERROR DETECTED??
118 057646  001403                  BEQ     21$              ;NO!!
119 057650  062716  000004          ADD     #4,(SP)          ;YES - MOVE SP TO ERROR RETURN
120 057654  000402                  BR      23$
121 057656  162716  000004  21$:    SUB     #4,(SP) ;MOVE SP BACK TO NO ERROR RETURN
122 057662  000240          23$:    NOP
123 057664  000207                  RTS     PC               ;RETURN TO USER
124
```

```
                                .SBTTL  SEARCH STATUS CHECK SUBROUTINE

                                ;THIS SUBROUTINE VERIFIES THE RESULTS OF SEARCH OPERATIONS USING
                                ;STATUS STORED IN THE GET BUFFER AND TEST CONDITIONS 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 USER'S "ERROR" TRAP.

                                ;THE FOLLOWING CONDITIONS ARE CHECKED:
                                ;
                                ;       .ANY ERROR WHICH OCCURRED WHILE READING OR WRITING REMOTE
                                ;REGISTERS IS REPORTED, I.E., "MCPE" = 1 OR "PAR" = 1
                                ;
                                ;       ."IAE" STATUS IS CHECKED IN ACCORDANCE WITH THE VALUE DETERMINED
                                ;BY THE PROGRAM, WHICH IS BASED ON FORMAT AND ADDRESS.
                                ;
                                ;       ."OPI", IF SET, IS REPORTED AS 1) "OPI" DUE TO MOL = 0, OR 2)
                                ;"OPI" DUE TO ON CYLINDER LATCH.
                                ;
                                ;       ."IVC", IF SET, IS REPORTED AS 1)  'IVC" ERROR WITH VOLUME
                                ;VALID ZERO, OR 2) ERRONEOUS "IVC" ERROR WITH VOLUME VALID SET.
                                ;
                                ;       ."SKI" IS REPORTED IF SET
                                ;
                                ;       ."DVC" IS REPORTED IF SET
                                ;
                                ;       .AN ERROR IS REPORTED IF "MOL" = 0, OR "PIP" = 1, OR "ATA"   0,
                                ;OR "VV" = 0.
                                ;
                                ;CALL:
                                ;(1)    JSR     PC,SCHSTS
                                ;(2)    BR      ??                      RETURN HERE IF NO ERROR
                                ;(3)    NOP                             RETURN HERE TO REPORT ERROR
                                ;(4)    ERROR                           SUBROUTINE WILL LOAD ERROR #
                                ;(5)    JSR     PC,@(SP)+               GO BACK FOR MORE CHECKS
                                ;(6)    ??                              RETURN AFTER ALL ERRORS REPORTED

057666                          SCHSTS:

                                ;CLEAR USER'S ERROR CALL
057666  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
057672  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
057676  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR BR
057702  005037  061230                  CLR     200$            ;CLEAR STATUS FLAGS

                                ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN WRITING REMOTE
                                ;REGISTERS, I.E., "PAR" = 1 AND "DPE" = 0.
057706  032737  000010  001350          BIT     #PAR,RMER11     ;WAS PARITY ERROR DETECTED??
057714  001431                  BEQ     10$             ;NO!!
057716  032737  000010  001376          BIT     #DPE,RMER21     ;WAS IT CONTROL BUS ERROR??
057724  001025                  BNE     10$             ;PROBABLY NOT!!

                                ;REPORT CONTROL BUS PARITY ERROR VIA USER'S ERROR CALL
057726  013737  001350  001140          MOV     RMER11,$GDDAT   ;EXPECTED STATUS
057734  042737  000010  001140          BIC     #PAR,$GDDAT
```

```
 58 057742  013737  001350  001142          MOV     RMER11,$BDDAT   ;RECEIVED STATUS
 59 057750  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 60 057754  112776  000050  000000          MOVB    #50,@(SP)       ;WRITE ERROR NUMBER IN CALL
 61 057762  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 62 057766  004736                          JSR     PC,@(SP)+       ;REPORT ERROR
 63 057770  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
 64 057774  000240                          NOP
 65 057776  000430                          BR      15$             ;SKIP FURTHER ERROR CHECKS
 66 060000                          10$:
 67
 68                                 ;TEST FOR MASSBUS CONTROL BUS PARITY ERROR WHEN READING REMOTE
 69                                 ;REGISTERS, I.E., "MCPE" = 1.
 70 060000  032737  020000  001334          BIT     #MCPE,RMCS11    ;WAS PARITY ERROR DETECTED??
 71 060006  001426                          BEQ     20$             ;NO.!
 72
 73                                 ;REPORT CONTROL BUS PARITY ERROR VIA USER'S ERROR CALL.
 74 060010  013737  001334  001140          MOV     RMCS11,$GDDAT   ;EXPECTED STATUS
 75 060016  042737  020000  001140          BIC     #MCPE,$GDDAT
 76 060024  013737  001334  001142          MOV     RMCS11,$BDDAT   ;RECEIVED STATUS
 77 060032  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
 78 060036  112776  000013  000000          MOVB    #13,@(SP)       ;WRITE ERROR NUMBER
 79 060044  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
 80 060050  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
 81 060052  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
 82 060056  000240                          NOP
 83 060060  000137  061202          15$:    JMP     150$            ;OMIT STATUS CHECKING
 84 060064                          20$:
 85
 86                                 ;DETERMINE THE VALUE OF "IAE" STATUS BASED ON TRACK, SECTOR AND CYLINDER
 87                                 ;ALSO SET "SKI" FLAG IF CYLINDER ADDRESS IS TOO LARGE.
 88 060064  012737  002000  001140          MOV     #IAE,$GDDAT     ;SETUP FOR IAE = 1
 89 060072  052737  040000  061230          BIS     #SKI,200$       ;SETUP FOR SKI = 1
 90 060100  023727  001444  001466          CMP     RMDCO,#822.     ;GREATER THAN LAST CYLINDER ?
 91 060106  101025                          BHI     30$             ;YES - CYLINDER IS INVALID
 92 060110  042737  040000  061230          BIC     #SKI,200$       ;"SKI" SHOULD BE ZERO
 93
 94 060116  123737  001417  001333          CMPB    RMDAO+1,LSTRK+1 ;GREATER THAN LAST TRACK ?
 95 060124  101016                          BHI     30$             ;YES - TRACK IS INVALID
 96
 97 060126  123727  001416  000035          CMPB    RMDAO,#29.      ;SECTOR > 29. ?
 98 060134  101410                          BLOS    25$             ;NO!!
 99 060136  032737  010000  001442          BIT     #FMT16,RMOFO    ;18 BIT FORMAT ?
100 060144  001406                          BEQ     30$             ;YES - SECTOR IS INVALID FOR 18 BIT MODE
101 060146  123727  001416  000037          CMPB    RMDAO,#31.      ;SECTOR > 31. ?
102 060154  101002                          BHI     30$             ;YES - SECTOR IS INVALID
103
104 060156  005037  001140          25$:    CLR     $GDDAT          ;"IAE" SHOULD = 0
105
106                                ;COMPARE EXPECTED AND RECIVED "IAE" STATUS
107 060162  013737  001350  001142  30$:    MOV     RMER11,$BDDAT   ;GET RECEIVED IAE
108 060170  042737  175777  001142          BIC     #^CIAE,$BDDAT
109 060176  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS IAE CORRECT??
110 060204  001004                          BNE     40$             ;NO!!
111 060206  042737  040000  061230          BIC     #SKI,200$       ;SKI SHOULD BE ZERO
112 060214  000413                          BR      50$
113 060216                         40$:
114
```

```
                                        ;REPORT INCORRECT IAE STATUS VIA USER'S ERROR CALL
060216  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060222  112776  000257  000000                MOVB    #257,@(SP)      ;WRITE ERROR NUMBER IN CALL
060230  162716  000002                        SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060234  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060236  162716  000010                        SUB     #10,(SP)        ;RETURN SP TO NO ERROR
060242  000240                                NOP
060244                                  50$:

                                        ;SEE IF "SKI" OR "DVC" OR "IVC" IS SET
060244  032737  050200  001376                BIT     #SKI!DVC!IVC,RMER2I     ;ARE ANY BITS SET??
060252  001531                                BEQ     90$             ;NO!!

                                        ;REPORT ERROR IF "SKI" IS SET AND "SKI" FLAG IS NOT SET
060254  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED "SKI" STATUS
060262  042737  137777  001142                BIC     #^CSKI,$BDDAT
060270  013737  061230  001140                MOV     200$,$GDDAT     ;EXPECTED "SKI" STATUS
060276  042737  137777  001140                BIC     #^CSKI,$GDDAT
060304  023737  001140  001142                CMP     $GDDAT,$BDDAT   ;IS "SKI" OK??
060312  001422                                BEQ     60$             ;YES-CHANGE ERROR NUMBER
060314  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060320  112776  000263  000000                MOVB    #263,@(SP)      ;WRITE ERROR NUMBER
060326  032737  040000  001140                BIT     #SKI,$GDDAT     ;SHOULD "SKI" BE SET??
060334  001403                                BEQ     55$             ;NO!!
060336  112776  000267  000000                MOVB    #267,@(SP)      ;YES-CHANGE ERROR NUMBER
060344  162716  000002                55$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060350  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060352  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
060356  000240                                NOP
060360                                  60$:

                                        ;REPORT "IVC" ERROR AS
                                        ;       .IVC DUE TO LOSS OF VOLUME VALID
                                        ;       .ERRONEOUS IVC WITH VOLUME VALID SET
060360  032737  010000  001376                BIT     #IVC,RMER2I     ;IS IVC SET??
060366  001433                                BEQ     80$             ;NO!!
060370  013737  001376  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
060376  042737  010000  001140                BIC     #IVC,$GDDAT
060404  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
060412  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
060416  112776  000264  000000                MOVB    #264,@(SP)      ;WRITE ERROR NUMBER IN CALL
060424  032737  000100  001346                BIT     #VV,RMDSI       ;WAS VOLUME VALID??
060432  001403                                BEQ     70$             ;NO!!
060434  112776  000265  000000                MOVB    #265,@(SP)      ;YES - CHANGE ERROR NUMBER
060442  162716  000002                70$:    SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
060446  004736                                JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
060450  162716  000010                        SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
060454  000240                                NOP
060456                                  80$:

                                        ;REPORT ANY DEVICE FAULT, I.E., "DVC" = 1
060456  032737  000200  001376                BIT     #DVC,RMER2I     ;WAS THERE A DEVICE FAULT??
060464  001424                                BEQ     90$             ;NO!!
060466  013737  001376  001140                MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
060474  042737  000200  001140                BIC     #DVC,$GDDAT
060502  013737  001376  001142                MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
060510  062716  000004                        ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
```

```
    172 060514  112776  000266  000000          MOVB    #266,@(SP)      ;WRITE ERROR NUMBER IN CALL
    173 060522  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    174 060526  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    175 060530  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    176 060534  000240                          NOP
    177 060536                          90$:
    178
    179                                 ;REPORT ANY "OPI" ERROR AS
    180                                 ;       ."OPI" BECAUSE MEDIUM IS NOT ONLINE
    181                                 ;       ."OPI" BECAUSE "ON CYLINDER" DIDN'T DROP
    182 060536  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI SET??
    183 060544  001433                          BEQ     110$            ;NO!!
    184 060546  013737  001350  001140          MOV     RMER1I,$GDDAT   ;EXPECTED STATUS
    185 060554  042737  020000  001140          BIC     #OPI,$GDDAT
    186 060562  013737  001350  001142          MOV     RMER1I,$BDDAT   ;RECEIVED STATUS
    187 060570  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    188 060574  112776  000270  000000          MOVB    #270,@(SP)      ;SETUP ERROR FOR MOL = 0
    189 060602  032737  010000  001346          BIT     #MOL,RMDSI      ;WAS MOL 0??
    190 060610  001403                          BEQ     105$            ;YES!!
    191 060612  112776  000271  000000          MOVB    #271,@(SP)      ;MOL WAS 1 - CHANGE ERROR
    192 060620  162716  000002          105$:   SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    193 060624  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    194 060626  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    195 060632  000240                          NOP
    196 060634                          110$:
    197
    198                                 ;SEE IF "ATA" = "MOL" = "VV" = 1 AND "PIP" = 0
    199 060634  013746  001346                  MOV     RMDSI,-(SP)     ;GET DRIVE STATUS
    200 060640  042716  047677                  BIC     #^C<ATA!PIP!MOL.VV>,(SP)
    201 060644  022726  110100                  CMP     #ATA!MOL!VV,(SP)+       ;IS DRIVE STATUS CORRECT??
    202 060650  001554                          BEQ     150$            ;YES!!
    203
    204                                 ;REPORT AN ERROR IF MOL = 0 AND OPI ERROR WAS NOT REPORTED, I.E., OPI - 0
    205 060652  032737  010000  001346          BIT     #MOL,RMDSI      ;WAS MEDIUM OFF LINE??
    206 060660  001030                          BNE     120$            ;NO!!
    207 060662  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
    208 060670  052737  010000  001140          BIS     #MOL,$GDDAT
    209 060676  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
    210 060704  032737  020000  001350          BIT     #OPI,RMER1I     ;WAS OPI REPORTED BEFORE??
    211 060712  001013                          BNE     120$            ;YES - DON'T REPORT MOL = 0
    212 060714  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    213 060720  112776  000272  000000          MOVB    #272,@(SP)      ;WRITE ERROR NUMBER IN CALL
    214 060726  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
    215 060732  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
    216 060734  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
    217 060740  000240                          NOP
    218 060742                          120$:
    219
    220                                 ;REPORT AN ERROR IF PIP IS STIL SET AND SKI IS RESET
    221 060742  032737  020000  001346          BIT     #PIP,RMDSI      ;IS POSITIONING IN PROGRESS??
    222 060750  001430                          BEQ     130$            ;NO!!
    223 060752  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS "SKI" DETECTED??
    224 060760  001024                          BNE     130$            ;YES-DONT REPORT PIP
    225 060762  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
    226 060770  042737  020000  001140          BIC     #PIP,$GDDAT
    227 060776  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
    228 061004  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
```

```
061010  112776  000273  000000          MOVB    #273,@(SP)      ;WRITE ERROR NUMBER IN CALL
061016  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061022  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061024  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
061030  000240                          NOP
061032                          130$:

                                ;REPORT AN ERROR IF VOLUME IS NOT VALID AND IVC = 0
061032  032737  000100  001346          BIT     #VV,RMDSI       ;IS VOLUME VALID??
061040  001030                          BNE     140$            ;YES!!
061042  032737  010000  001376          BIT     #IVC,RMER2I     ;WAS IVC DETECTED??
061050  001024                          BNE     140$            ;YES - DON'T REPORT VV = 0
061052  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
061060  052737  000100  001140          BIS     #VV,$GDDAT
061066  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
061074  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USERS ERROR CALL
061100  112776  000350  000000          MOVB    #350,@(SP)      ;WRITE ERROR NUMBER IN CALL
061106  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061112  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061114  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
061120  000240                          NOP
061122                          140$:

                                ;REPORT AN ERROR IF ATTENTION IS NOT SET
061122  032737  100000  001346          BIT     #ATA,RMDSI      ;IS ATA ON??
061130  001024                          BNE     150$            ;YES!!
061132  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
061140  052737  100000  001140          BIS     #ATA,$GDDAT
061146  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
061154  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
061160  112776  000351  000000          MOVB    #351,@(SP)      ;WRITE ERROR NUMBER IN CALL
061166  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN IF ERROR
061172  004736                          JSR     PC,@(SP)+       ;REPORT ERROR AND RETURN
061174  162716  000010                  SUB     #10,(SP)        ;RESTORE SP TO NO ERROR
061200  000240                          NOP
061202                          150$:

                                ;ARGUMENT THE RETURN ADDRESS IF AN ERROR WAS DETECTED
061202  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
061206  105776  000000                  TSTB    @(SP)           ;WAS ERROR FOUND??
061212  001403                          BEQ     160$            ;NO!!
061214  062716  000004                  ADD     #4,(SP)         ;YES - CHANGE RETURN
061220  000402                          BR      170$
061222  162716  000004          160$:   SUB     #4,(SP)         ;NO ERROR FOUND
061226  000207                  170$:   RTS     PC              ;RETURN TO USER

061230  000000                  200$:   .WORD                   ;STORAGE FOR FLAGS
```

```
                                .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 REPOR1 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

061232                          STCDRVSTS:

                                ;CLEAR USER'S ERROR CALL
061232  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
061236  105076  000000                  CLRB    @(SP)           ;CLEAR ERROR NUMBER
061242  162716  000004                  SUB     #4,(SP)         ;MOVE SP BACK TO NO ERROR RETURN
                                ;SEE IF "MOL" = "VV" = 1, AND "PIP" = 0
061246  013746  001346                  MOV     RMDSI,-(SP)     ;PUT DRIVE STATUS ON STACK
061252  042716  147677                  BIC     #^C<PIP!MOL!VV>,(SP)
061256  022726  010100                  CMP     #MOL!VV,(SP)+   ;ARE MOL,VV AND PIP O.K.??
061262  001524                          BEQ     30$             ;YES!!

                                ;REPORT AN ERROR IF MOL = 0 AND "OPI" = 0
061264  032737  010000  001346          BIT     #MOL,RMDSI      ;IS MOL ON ??
061272  001030                          BNE     10$             ;YES!!
061274  032737  020000  001340          BIT     #OPI,RMER1I     ;WAS "OPI" SET??
061302  001024                          BNE     10$             ;YES-DONT REPORT "MOL" - 0
061304  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
061312  052737  010000  001140          BIS     #MOL,$GDDAT
061320  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
061326  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
061332  112776  000207  000000          MOVB    #207,@(SP)      ;WRITE ERROR NUMBER IN CALL
061340  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
061344  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
061346  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR RETURN
061352  000240                          NOP
061354                          10$:

                                ;REPORT AN ERROR IF VOLUME VALID IS NOW ZERO AND "IVC"   0
061354  032737  000100  001346          BIT     #VV,RMDSI       ;IS "VV" - 0??
061362  001030                          BNE     20$             ;NO!.
```

```
     58 061364  032737  010000  001376          BIT     #IVC,RMER2I     ;WAS "IVC" SET??
     59 061372  001024                          BNE     20$             ;YES-DONT REPORT "VV" = 0
     60 061374  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
     61 061402  052737  000100  001346          BIS     #VV,RMDSI
     62 061410  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
     63 061416  062716  000004                  ADD     #4,(SP) ;MOVE SP TO USER'S ERROR CALL
     64 061422  112776  000210  000000          MOVB    #210,@(SP)      ;WRITE ERROR NUMBER IN CALL
     65 061430  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     66 061434  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     67 061436  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
     68 061442  000240                          NOP
     69 061444                          20$:
     70
     71                                 ;REPORT AN ERROR IF DRIVE IS OFF CYLINDER AND "SKI" = 0
     72 061444  032737  020000  001346          BIT     #PIP,RMDSI      ;IS DRIVE OFF CYLINDER??
     73 061452  001430                          BEQ     30$             ;NO!!
     74 061454  032737  040000  001376          BIT     #SKI,RMER2I     ;WAS "SKI" SET??
     75 061462  001024                          BNE     30$             ;YES-DONT REPORT "PIP" = 1
     76 061464  013737  001346  001140          MOV     RMDSI,$GDDAT    ;EXPECTED STATUS
     77 061472  042737  020000  001140          BIC     #PIP,$GDDAT
     78 061500  013737  001346  001142          MOV     RMDSI,$BDDAT    ;RECEIVED STATUS
     79 061506  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
     80 061512  112776  000211  000000          MOVB    #211,@(SP)      ;WRITE ERROR NUMBER IN USER'S CALL
     81 061520  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
     82 061524  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
     83 061526  162716  000010                  SUB     #10,(SP)        ;MOVE SP TO NO ERROR RETURN
     84 061532  000240                          NOP
     85 061534                          30$:
     86
     87                                 ;SEE IF "SKI" = "DVC" = 0
     88 061534  013746  001376                  MOV     RMER2I,-(SP)    ;PUT ERROR REG 2 ON STACK
     89 061540  042726  137577                  BIC     #^C<SKI!DVC>,(SP)+
     90 061544  001460                          BEQ     60$             ;BRANCH IF NO ERROR
     91 061546                          40$:
     92
     93                                 ;REPORT AN ERROR IF THERE IS A DEVICE FAULT
     94 061546  032737  000200  001376          BIT     #DVC,RMER2I     ;ANY DEVICE FAULT??
     95 061554  001424                          BEQ     50$             ;NO!!
     96 061556  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
     97 061564  042737  000200  001140          BIC     #DVC,$GDDAT
     98 061572  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
     99 061600  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S CALL
    100 061604  112776  000212  000000          MOVB    #212,@(SP)      ;WRITE NUMBER OF ERROR IN CALL
    101 061612  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    102 061616  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    103 061620  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    104 061624  000240                          NOP
    105 061626                          50$:
    106
    107                                 ;REPORT AN ERROR IF "SKI" = 1
    108 061626  032737  040000  001376          BIT     #SKI,RMER2I     ;IS THERE A SEEK INCOMPLETE ERROR
    109 061634  001424                          BEQ     60$             ;NO!!
    110 061636  013737  001376  001140          MOV     RMER2I,$GDDAT   ;EXPECTED STATUS
    111 061644  042737  040000  001140          BIC     #SKI,$GDDAT
    112 061652  013737  001376  001142          MOV     RMER2I,$BDDAT   ;RECEIVED STATUS
    113 061660  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    114 061664  112776  000213  000000          MOVB    #213,@(SP)      ;WRITE ERROR NUMBER IN USER'S ERROR CALL
```

```
    115 061672  162716  000002                  SUB     #2,(SP)         ;MOVE SP TO RETURN FOR ERROR
    116 061676  004736                          JSR     PC,@(SP)+       ;REPORT ERROR VIA USER
    117 061700  162716  000010                  SUB     #10,(SP)        ;MOVE SP BACK TO NO ERROR
    118 061704  000240                          NOP
    119 061706                          60$:
    120
    121                                 ;AUGMENT THE RETURN ADDRESS IF ANY ERROR WAS DETECTED
    122 061706  062716  000004                  ADD     #4,(SP)         ;MOVE SP TO USER'S ERROR CALL
    123 061712  105776  000000                  TSTB    @(SP)           ;WAS AN ERROR DETECTED??
    124 061716  001403                          BEQ     70$             ;NO...
    125 061720  062716  000004                  ADD     #4,(SP)         ;YES - MOVE SP TO USER'S ERROR RETURN
    126 061724  000402                          BR      80$
    127 061726  162716  000004          70$:    SUB     #4,(SP)         ;NO - MOVE SP TO NO ERROR RETURN
    128 061732  000240                  80$:    NOP
    129 061734  000207                          RTS     PC              ;RETURN TO USER
    130
```

```
                        .SBTTL  STOP AND SHUTDOWN SUBROUTINES

                        STOP:

                        ;DROP PRIORITY TO ALLOW CONSOLE INTERRUPT
061736  012746  000140          MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
061742  012746  061750          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
061746  000002                  RTI                     ;;POP NEW PC AND PS
061750                  64$:
061750  000240                  NOP

                        ;RAISE PRIORITY TO INHIBIT CONSOLE INTERRUPT
061752  012746  000300          MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
061756  012746  061764          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
061762  000002                  RTI                     ;;POP NEW PC AND PS
061764                  65$:
061764  000207                  RTS     PC              ;CONTINUE

061766  005737  000042  SHUT:   TST     @#42            ;ANY MONITOR PRESENT ?
061772  001015                  BNE     10$             ;BR IF YES
061774  104401  062002          TYPE    ,65$            ;;TYPE ASCIZ STRING
062000  000410                  BR      64$             ;;GET OVER THE ASCIZ
                        ;;65$:  .ASCIZ  <CRLF><07>/TEST HALTED/<CRLF>
062022                  64$:
062022  000137  005412          JMP     START           ;GO TO START
062026  000137  042136  10$:    JMP     $EOP            ;RETURN CONTROL TO MONITOR
```

```
      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

   062032                             $SAVREG:
   062032  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
   062034  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   062036  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   062040  010346                             MOV     R3,-(SP)        ;;PUSH R3 ON STACK
   062042  010446                             MOV     R4,-(SP)        ;;PUSH R4 ON STACK
   062044  010546                             MOV     R5,-(SP)        ;;PUSH R5 ON STACK
   062046  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
   062052  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
   062056  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
   062062  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
   062066  000002                             RTI

                                      ;*RESTORE R0-R5
                                      ;*CALL:
                                      ;*      RESREG
   062070                             $RESREG:
   062070  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
   062074  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
   062100  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
   062104  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
   062110  012605                             MOV     (SP)+,R5        ;;POP STACK INTO R5
   062112  012604                             MOV     (SP)+,R4        ;;POP STACK INTO R4
   062114  012603                             MOV     (SP)+,R3        ;;POP STACK INTO R3
   062116  012602                             MOV     (SP)+,R2        ;;POP STACK INTO R2
   062120  012601                             MOV     (SP)+,R1        ;;POP STACK INTO R1
   062122  012600                             MOV     (SP)+,R0        ;;POP STACK INTO R0
   062124  000002                             RTI
      2                               .SBTTL  BINARY TO ASCII AND TYPE ROUTINE

                                      ;;*****************************************************************
                                      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
                                      ;*BINARY-ASCII NUMBER AND TYPE IT.
                                      ;*CALL:
                                      ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
                                      ;*      TYPBN                   ;;TYPE IT

   062126  010146                     $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
   062130  016601  000006                     MOV     6(SP),R1        ;;GET THE INPUT NUMBER
   062134  000261                             SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
```

```
062136  112737  000060  062200  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
062144  006101                          ROL     R1              ;;GET THIS BIT
062146  001406                          BEQ     2$              ;;DONE?
062150  105537  062200                  ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
062154  104401  062200                  TYPE    ,$BIN           ;;GO TYPE THIS BIT
062160  000241                          CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
062162  000765                          BR      1$              ;;GO DO THE NEXT BIT
062164  012601                  2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
062166  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
062174  012616                          MOV     (SP)+,(SP)
062176  000002                          RTI                     ;;RETURN TO USER
062200     000     000          $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
      3                         .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                   ;;GO TO THE ROUTINE

062202                          $TYPDS:
062202  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
062204  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
062206  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
062210  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
062212  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
062214  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
062220  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
062224  100004                          BPL     1$              ;;BR IF INPUT IS POS.
062226  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
062230  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
062236  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
062240  012703  062416                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
062244  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
062250  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
062252  016001  062406                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
062256  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
062260  002402                          BLT     4$              ;;BR IF DONE
062262  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
062264  000774                          BR      3$
062266  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
062270  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
062272  001002                          BNE     5$              ;;FALL THROUGH IF 0
062274  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
062276  100407                          BMI     7$              ;;BR IF YES
062300  106316                  5$:     ASLB    (SP)            ;;MSD?
062302  103003                          BCC     6$              ;;BR IF NO
062304  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
062312  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
062316  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
062322  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
062324  005720                          TST     (R0)+           ;;JUST INCREMENTING
062326  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
```

```
062332  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
062334  003002                          BGT     8$              ;;GO TO EXIT
062336  010502                          MOV     R5,R2           ;;GET THE LSD
062340  000764                          BR      6$              ;;GO CHANGE TO ASCII
062342  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
062344  100003                          BPL     9$              ;;BR IF NO
062346  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
062354  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
062356  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
062360  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
062362  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
062364  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
062366  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
062370  104401  062416                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
062374  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
062402  012616                          MOV     (SP)+,(SP)
062404  000002                          RTI                     ;;RETURN TO USER
062406  023420                  $DTBL:  10000.
062410  001750                          1000.
062412  000144                          100.
062414  000012                          10.
062416                          $DBLK:  .BLKW   4
    4                           .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;**********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

062426  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
062432  116637  000001  062651          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
062440  112637  062653                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
062444  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
062450  000406                          BR      $TYPON
062452  112737  000001  062651  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
062460  112737  000006  062653          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
062466  112737  000005  062650  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
062474  010346                          MOV     R3,-(SP)        ;;SAVE R3
062476  010446                          MOV     R4,-(SP)        ;;SAVE R4
```

```
062500  010546                          MOV     R5,-(SP)        ;;SAVE R5
062502  113704  062653                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
062506  005404                          NEG     R4
062510  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
062514  110437  062652                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
062520  113704  062651                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
062524  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
062530  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
062532  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
062534  000404                          BR      3$              ;;GO DO MSB
062536  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
062540  006105                          ROL     R5
062542  006105                          ROL     R5
062544  010503                          MOV     R5,R3
062546  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
062550  105337  062652                  DECB    $OMODE          ;;TYPE THIS DIGIT?
062554  100016                          BPL     7$              ;;BR IF NO
062556  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
062562  001002                          BNE     4$              ;;TEST FOR 0
062564  005704                          TST     R4              ;;SUPPRESS THIS 0?
062566  001403                          BEQ     5$              ;;BR IF YES
062570  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
062572  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
062576  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
062602  110337  062646                  MOVB    R3,8$           ;;SAVE FOR TYPING
062606  104401  062646                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
062612  105337  062650          7$:     DECB    $OCNT           ;;COUNT BY 1
062616  003347                          BGT     2$              ;;BR IF MORE TO DO
062620  002402                          BLT     6$              ;;BR IF DONE
062622  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
062624  000744                          BR      2$              ;;GO DO THE LAST DIGIT
062626  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
062630  012604                          MOV     (SP)+,R4        ;;RESTORE R4
062632  012603                          MOV     (SP)+,R3        ;;RESTORE R3
062634  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
062642  012616                          MOV     (SP)+,(SP)
062644  000002                          RTI                     ;;RETURN
062646     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
062647     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
062650     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
062651     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
062652  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5                               .SBTTL  TYPE ROUTINE

                        ;;**********************************************************************
                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                        ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                        ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                        ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                        ;*
                        ;*CALL:
                        ;*1) USING A TRAP INSTRUCTION
                        ;*       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                        ;*OR
                        ;*       TYPE
                        ;*       MESADR
```

```
                                ;*
062654  105737  001173          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
062660  100002                          BPL     1$              ;;BR IF YES
062662  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
062664  000430                          BR      3$              ;;LEAVE
062666  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
062670  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
062674  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
062702  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
062704  132737  000100  001243          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
062712  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
062714  010037  062724                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
062720  004737  065742                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
062724  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
062726  132737  000040  001243  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
062734  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
062736  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
062740  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
062742  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
062744  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
062746  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
062752  000002                          RTI                     ;;RETURN
062754  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
062760  001430                          BEQ     8$
062762  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
062766  001006                          BNE     5$
062770  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
062772  104401                          TYPE                    ;;TYPE A CR AND LF
062774  001217                          $CRLF
062776  105037  063204                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
063002  000755                          BR      2$              ;;GET NEXT CHARACTER
063004  004737  063066          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
063010  123726  001172          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
063014  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
063016  013746  001170                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
063022  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
063026  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
063030  004737  063066                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
063034  105337  063204                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
063040  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

063042  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
063046  004737  063066          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
063052  132737  000007  063204          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
063060  001372                          BNE     9$              ;;TAB STOP
063062  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
063064  000724                          BR      2$              ;;GET NEXT CHARACTER
063066                          $TYPEC:
063066  105777  116066                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
063072  100022                          BPL     10$             ;;BR IF NO!
063074  017746  116062                  MOV     @$TKB,-(SP)     ;;GET CHAR
063100  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
063104  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
```

```
063110  001012                          BNE     102$            ;;BR IF NOT
063112                          101$:
063112  105777  116042                  TSTB    @$TKS           ;;WAIT FOR CHAR
063116  100375                          BPL     101$
063120  117716  116036                  MOVB    @$TKB,(SP)      ;;GET CHAR
063124  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
063130  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
063134  001366                          BNE     101$            ;;BR IF NOT
063136                          102$:
063136  005726                          TST     (SP)+           ;;FIX STACK
063140                          10$:
063140  105777  116020                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
063144  100375                          BPL     10$
063146  116677  000002  116012          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
063154  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
063162  001003                          BNE     1$              ;;BRANCH IF NO
063164  105037  063204                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
063170  000406                          BR      $TYPEX          ;;EXIT
063172  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
063200  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
063202  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
063204  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
063206  000207                  $TYPEX: RTS     PC

  6                             .SBTTL  SCOPE HANDLER ROUTINE

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

063210                          $SCOPE:
063210  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
063212  004737  061736                  JSR     PC,STOP
063216  032777  040000  115730  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
063224  001131                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
063226  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
063230  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
063234  012737  063254  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
063242  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
063246  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
063252  000500                          BR      $SVLAD          ;;GO TO THE NEXT TEST
063254  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
063256  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
063262  000440                          BR      7$              ;;LOOP ON THE PRESENT TEST
063264                          6$:;#####END OF CODE FOR THE XOR TESTER#####
063264  032777  000400  115662          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
063272  001421                          BEQ     2$              ;;BR IF NO
```

```
063274  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
063276  117716  115652                  MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
063302  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
063304  022716  000071                  CMP     #71,(SP)        ;;CHECK THE NUMBER IN THE SWR
063310  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
063312  011637  001116                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
063316  005316                          DEC     (SP)            ;;BACKUP BY ONE
063320  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
063322  062716  063526                  ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
063326  013637  001122                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
063332  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
063334  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
063336  105737  001117          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
063342  001421                          BEQ     3$              ;;BR IF NO
063344  123737  001131  001117          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
063352  101015                          BHI     3$              ;;BR IF NO
063354  032777  001000  115572          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
063362  001404                          BEQ     4$              ;;BR IF NO
063364  013737  001124  001122  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
063372  000446                          BR      $OVER
063374  105037  001117          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
063400  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
063404  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
063406  032777  004000  115540  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
063414  001011                          BNE     1$              ;;BR IF YES
063416  005737  001230                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
063422  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
063424  005237  001120                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
063430  023737  001206  001120          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
063436  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
063440  012737  000001  001120  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
063446  013737  063524  001206          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
063454  105237  001116          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
063460  113737  001116  001226          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
063466  011637  001122                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
063472  011637  001124                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
063476  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
063502  112737  000001  001131          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
063510  013777  001116  115440  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
063516  013716  001122                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
063522  000002                          RTI                     ;;FIXES PS
063524  000012                  $MXCNT: 10.                     ;;MAX. NUMBER OF ITERATIONS
063526                          $SW08TBL:
063526  010120                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
063530  010304                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
063532  010470                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
063534  010672                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
063536  011542                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
063540  012100                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
063542  012304                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
063544  012620                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
063546  013000                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
063550  014052                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
063552  014352                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
063554  014570                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
063556  015076                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
063560  015406                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
```

```
063562  015650                  .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
063564  016126                  .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
063566  016376                  .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
063570  016626                  .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
063572  017152                  .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
063574  017572                  .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
063576  020136                  .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
063600  020530                  .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
063602  021220                  .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
063604  021566                  .WORD   TST30+2         ;;STARTING ADDRESS OF TEST 30
063606  022110                  .WORD   TST31+2         ;;STARTING ADDRESS OF TEST 31
063610  022472                  .WORD   TST32+2         ;;STARTING ADDRESS OF TEST 32
063612  023010                  .WORD   TST33+2         ;;STARTING ADDRESS OF TEST 33
063614  023574                  .WORD   TST34+2         ;;STARTING ADDRESS OF TEST 34
063616  024320                  .WORD   TST35+2         ;;STARTING ADDRESS OF TEST 35
063620  025110                  .WORD   TST36+2         ;;STARTING ADDRESS OF TEST 36
063622  025674                  .WORD   TST37+2         ;;STARTING ADDRESS OF TEST 37
063624  026226                  .WORD   TST40+2         ;;STARTING ADDRESS OF TEST 40
063626  026666                  .WORD   TST41+2         ;;STARTING ADDRESS OF TEST 41
063630  027106                  .WORD   TST42+2         ;;STARTING ADDRESS OF TEST 42
063632  027444                  .WORD   TST43+2         ;;STARTING ADDRESS OF TEST 43
063634  027732                  .WORD   TST44+2         ;;STARTING ADDRESS OF TEST 44
063636  030220                  .WORD   TST45+2         ;;STARTING ADDRESS OF TEST 45
063640  030532                  .WORD   TST46+2         ;;STARTING ADDRESS OF TEST 46
063642  031040                  .WORD   TST47+2         ;;STARTING ADDRESS OF TEST 47
063644  031352                  .WORD   TST50+2         ;;STARTING ADDRESS OF TEST 50
063646  031662                  .WORD   TST51+2         ;;STARTING ADDRESS OF TEST 51
063650  032174                  .WORD   TST52+2         ;;STARTING ADDRESS OF TEST 52
063652  032612                  .WORD   TST53+2         ;;STARTING ADDRESS OF TEST 53
063654  033240                  .WORD   TST54+2         ;;STARTING ADDRESS OF TEST 54
063656  033656                  .WORD   TST55+2         ;;STARTING ADDRESS OF TEST 55
063660  034216                  .WORD   TST56+2         ;;STARTING ADDRESS OF TEST 56
063662  034560                  .WORD   TST57+2         ;;STARTING ADDRESS OF TEST 57
063664  035112                  .WORD   TST60+2         ;;STARTING ADDRESS OF TEST 60
063666  035540                  .WORD   TST61+2         ;;STARTING ADDRESS OF TEST 61
063670  036216                  .WORD   TST62+2         ;;STARTING ADDRESS OF TEST 62
063672  036722                  .WORD   TST63+2         ;;STARTING ADDRESS OF TEST 63
063674  037322                  .WORD   TST64+2         ;;STARTING ADDRESS OF TEST 64
063676  037732                  .WORD   TST65+2         ;;STARTING ADDRESS OF TEST 65
063700  040332                  .WORD   TST66+2         ;;STARTING ADDRESS OF TEST 66
063702  040712                  .WORD   TST67+2         ;;STARTING ADDRESS OF TEST 67
063704  041256                  .WORD   TST70+2         ;;STARTING ADDRESS OF TEST 70
063706  041614                  .WORD   TST71+2         ;;STARTING ADDRESS OF TEST 71

.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
```

```
063710                          $ERROR:
063710  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
063712  105237  001117          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
063716  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
063720  013777  001116  115230          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
063726  032777  002000  115220          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
063734  001402                          BEQ     1$              ;;NO - SKIP
063736  104401  001212                  TYPE    ,$BELL          ;;RING BELL
063742  005237  001126          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
063746  011637  001132                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
063752  162737  000002  001132          SUB     #2,$ERRPC
063760  117737  115146  001130          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
063766  032777  020000  115160          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
063774  001004                          BNE     20$             ;;SKIP TYPEOUTS
063776  004737  042346                  JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
064002  104401  001217                  TYPE    ,$CRLF
064006                          20$:
064006  122737  000001  001242          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
064014  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
064016  113737  001130  064030          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
064024  004737  065752                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
064030     000          21$:    .BYTE   0
064031     000                  .BYTE   0
064032  000777          22$:    BR      22$             ;;APT ERROR LOOP
064034  005777  115114          2$:     TST     @SWR            ;;HALT ON ERROR
064040  100002                          BPL     3$              ;;SKIP IF CONTINUE
064042  000000                          HALT                    ;;HALT ON ERROR!
064044  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
064046  032777  001000  115100  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
064054  001402                          BEQ     4$              ;;BR IF NO
064056  013716  001124                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
064062  005737  001210          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
064066  001402                          BEQ     5$              ;;BR IF NONE
064070  013716  001210                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
064074                          5$:
064074  022737  042326  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
064102  001001                          BNE     6$              ;;BRANCH IF NO
064104  000000                          HALT                    ;;YES
064106                          6$:
064106  000002                          RTI                     ;;RETURN
8                               .SBTTL  TTY INPUT ROUTINE

                                ;;**********************************************************************
                                .ENABL  LSB
064110  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
064112  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
064114  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
064116                          $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        064117                  $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
                                ;
064120  005037  064110          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
064124  012737  064116  064112          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
064132  013737  064112  064114          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
064140  012737  064170  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
064146  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
064154  005777  115002                  TST     @$TKB           ;;CLEAR DONE FLAG
064160  012777  000100  114772          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
064166  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 (SHUT)
                                ;
064170  117746  114766          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
064174  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
064200  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
064204  001007                          BNE     1$              ;;BRANCH IF NO
064206  104401  065304                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
064212  004737  064120                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
064216  005726                          TST     (SP)+           ;;CLEAN UP STACK
064220  000137  061766                  JMP     SHUT            ;;CONTROL C RESTART
064224  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
064230  001004                          BNE     2$              ;;BRANCH IF NO
064232  022737  000176  001154          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
064240  001500                          BEQ     6$              ;;GO TO SWR CHANGE
064242                          2$:
064242  022737  000001  064110          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
064250  001004                          BNE     3$              ;;BRANCH IF NO
064252  104401  001212                  TYPE    ,$BELL          ;;RING THE TTY BELL
064256  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
064260  000451                          BR      5$              ;;EXIT
064262  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
064266  001021                          BNE     32$             ;;BRANCH IF NO
064270  005077  114664                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
064274  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
064276  105777  114656          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
064302  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
064304  117746  114652                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
064310  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
064314  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
064320  001366                          BNE     31$             ;;BRANCH IF NO
064322  012777  000100  114630          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
064330  000002                          RTI                     ;;RETURN
064332  005237  064110          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
064336  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
064342  002405                          BLT     4$              ;;BRANCH IF YES
064344  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
064350  003002                          BGT     4$              ;;BRANCH IF YES
064352  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
064356  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
064362  005237  064112                  INC     $TKQIN          ;;UPDATE THE POINTER
```

```
064366  023727  064112  064117          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
064374  001003                          BNE     5$              ;;BRANCH IF NO
064376  012737  064116  064112          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
064404  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.
064406  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
064414  001124                          BNE     15$             ;;EXIT IF NOT
064416  105777  114536                  TSTB    @$TKS           ;;IS A CHAR WAITING?
064422  100121                          BPL     15$             ;;IF NOT, EXIT
064424  117746  114532                  MOVB    @$TKB,-(SP)     ;;YES
064430  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
064434  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
064440  001300                          BNE     2$              ;;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.
064442  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
064450  001674                          BEQ     2$              ;;BRANCH IF YES
064452  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
064454  004737  064120                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
064460  005077  114474                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
064464  112737  000001  001151          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

064472  104401  065316                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
064476  104401  065323          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
064502  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
064506  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
064510  104401  065334                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
064514  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
064516  005046                          CLR     -(SP)           ;;THE NEW SWR
064520  105777  114434          7$:     TSTB    @$TKS           ;;CHAR THERE?
064524  100375                          BPL     7$              ;;IF NOT TRY AGAIN

064526  117746  114430                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
064532  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

064536  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
064542  001015                          BNE     9$              ;;BRANCH IF NOT
064544  104401  065304                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
064550  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
064554  123727  001151  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
064562  001003                          BNE     8$              ;;BRANCH IF NO
064564  012777  000100  114366          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
064572  000137  061766          8$:     JMP     SHUT            ;;CONTROL-C RESTART


064576  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
064602  001005                          BNE     10$             ;;BRANCH IF NOT
064604  104401  065311                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
```

```
064610  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
064614  000737                          BR      19$             ;;LET'S TRY IT AGAIN


064616  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
064622  001022                          BNE     16$             ;;BRANCH IF NO
064624  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
064630  001403                          BEQ     11$             ;;BRANCH IF YES
064632  016677  000002  114314          MOV     2(SP),@SWR      ;;SAVE NEW SWR
064640  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
064644  104401  001217          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
064650  123727  001151  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
064656  001003                          BNE     15$             ;;BRANCH IF NOT
064660  012777  000100  114272          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
064666  000002          15$:    RTI                             ;;RETURN
064670  004737  063066          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
064674  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
064700  002420                          BLT     18$             ;;BRANCH IF YES
064702  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
064706  003015                          BGT     18$             ;;BRANCH IF YES
064710  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
064714  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
064720  001403                          BEQ     17$             ;;BRANCH IF YES
064722  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
064724  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
064726  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
064730  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
064734  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
064740  000667                          BR      7$              ;;GET THE NEXT ONE
064742  104401  001216          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
064746  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
                                ;

064750  011046          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
064752  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
064760  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
064764  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
064766  012746  064774                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
064772  000002                          RTI                     ;;POP NEW PC AND PS
064774                          64$:
064774  005737  064110          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
065000  001775                          BEQ     1$
065002  005337  064110                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
065006  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
065014  005237  064114                  INC     $TKQOUT         ;;UPDATE THE POINTER
065020  023727  064114  064117          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
065026  001003                          BNE     2$              ;;BRANCH IF NO
065030  012737  064116  064114          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
```

```
065036  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

065040  010346                  $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
065042  005046                          CLR     -(SP)                   ;;CLEAR THE RUBOUT KEY
065044  012703  065274            1$:     MOV     #$TTYIN,R3              ;;GET ADDRESS
065050  022703  065304            2$:     CMP     #$TTYIN+8.,R3           ;;BUFFER FULL?
065054  101456                          BLOS    4$                      ;;BR IF YES
065056  104411                          RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
065060  112613                          MOVB    (SP)+,(R3)              ;;GET CHARACTER
065062  122713  000177            10$:    CMPB    #177,(R3)               ;;IS IT A RUBOUT
065066  001022                          BNE     5$                      ;;BR IF NO
065070  005716                          TST     (SP)                    ;;IS THIS THE FIRST RUBOUT?
065072  001007                          BNE     6$                      ;;BR IF NO
065074  112737  000134  065272          MOVB    #'\,9$                  ;;TYPE A BACK SLASH
065102  104401  065272                  TYPE    ,9$
065106  012716  177777                  MOV     #-1,(SP)                ;;SET THE RUBOUT KEY
065112  005303                  6$:     DEC     R3                      ;;BACKUP BY ONE
065114  020327  065274                  CMP     R3,#$TTYIN              ;;STACK EMPTY?
065120  103434                          BLO     4$                      ;;BR IF YES
065122  111337  065272                  MOVB    (R3),9$                 ;;SETUP TO TYPEOUT THE DELETED CHAR.
065126  104401  065272                  TYPE    ,9$                     ;;GO TYPE
065132  000746                          BR      2$                      ;;GO READ ANOTHER CHAR.
065134  005716                  5$:     TST     (SP)                    ;;RUBOUT KEY SET?
065136  001406                          BEQ     7$                      ;;BR IF NO
065140  112737  000134  065272          MOVB    #'\,9$                  ;;TYPE A BACK SLASH
065146  104401  065272                  TYPE    ,9$
065152  005016                          CLR     (SP)                    ;;CLEAR THE RUBOUT KEY
065154  122713  000025            7$:     CMPB    #25,(R3)                ;;IS CHARACTER A CTRL U?
065160  001003                          BNE     8$                      ;;BR IF NO
065162  104401  065311                  TYPE    ,$CNTLU                 ;;TYPE A CONTROL "U"
065166  000726                          BR      1$                      ;;GO START OVER
065170  122713  000022            8$:     CMPB    #22,(R3)                ;;IS CHARACTER A "^R"?
065174  001011                          BNE     3$                      ;;BRANCH IF NO
065176  105013                          CLRB    (R3)                    ;;CLEAR THE CHARACTER
065200  104401  001217                  TYPE    ,$CRLF                  ;;TYPE A "CR" & "LF"
065204  104401  065274                  TYPE    ,$TTYIN                 ;;TYPE THE INPUT STRING
065210  000717                          BR      2$                      ;;GO PICKUP ANOTHER CHACTER
065212  104401  001216            4$:     TYPE    ,$QUES                  ;;TYPE A '?'
065216  000712                          BR      1$                      ;;CLEAR THE BUFFER AND LOOP
065220  111337  065272            3$:     MOVB    (R3),9$                 ;;ECHO THE CHARACTER
065224  104401  065272                  TYPE    ,9$
065230  122723  000015                  CMPB    #15,(R3)+               ;;CHECK FOR RETURN
065234  001305                          BNE     2$                      ;;LOOP IF NOT RETURN
065236  105063  177777                  CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
065242  104401  001220                  TYPE    ,$LF                    ;;TYPE A LINE FEED
065246  005726                          TST     (SP)+                   ;;CLEAN RUBOUT KEY FROM THE STACK
065250  012603                          MOV     (SP)+,R3                ;;RESTORE R3
065252  011646                          MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
065254  016666  000004  000002          MOV     4(SP),2(SP)             ;;      FIRST ASCII CHARACTER ON IT
065262  012766  065274  000004          MOV     #$TTYIN,4(SP)
065270  000002                          RTI                             ;;RETURN
```

```
065272  000                     9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
065273  000                             .BYTE   0               ;;TERMINATOR
065274                          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
065304  136  103  015          $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
065311  136  125  015          $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
065316  136  107  015          $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
065323  015  012  123          $MSWR:  .ASCIZ  <15><12>/SWR = /
065334  040  040  116          $MNEW:  .ASCIZ  /  NEW = /
                                .EVEN
 9                              .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                ;;******************************************************************
                                ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                ;*CHANGE IT TO BINARY.
                                ;*CALL:
                                ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT

065346  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
065350  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
065356  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
065360  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
065362  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
065364  104412                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
065366  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
065370  005001                          CLR     R1              ;;CLEAR DATA WORD
065372  005002                          CLR     R2
065374  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
065376  001412                          BEQ     3$              ;;IF ZERO GET OUT
065400  006301                          ASL     R1              ;;*2
065402  006102                          ROL     R2
065404  006301                          ASL     R1              ;;*4
065406  006102                          ROL     R2
065410  006301                          ASL     R1              ;;*8
065412  006102                          ROL     R2
065414  042716  177770                 BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
065420  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
065422  000764                          BR      2$              ;;LOOP
065424  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
065426  010166  000012                 MOV     R1,12(SP)       ;;SAVE THE RESULT
065432  010237  065446                 MOV     R2,$HIOCT
065436  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
065440  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
065442  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
065444  000002                          RTI                     ;;RETURN
065446  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
10                              .SBTTL  TRAP DECODER

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

065450  016646  000002          $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
065454  042716  000020                  BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
```

```
065460  012746  065466                  MOV     #1$,-(SP)       ;; T-BIT TRAPS
065464  000002                          RTI                     ;;SET THE NEW STATUS
065466  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
065470  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
065474  005740                          TST     -(R0)           ;;BACKUP BY 2
065476  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
065500  006300                          ASL     R0              ;;POSITION FOR INDEXING
065502  016000  065522                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
065506  000200                          RTS     R0              ;;GO TO ROUTINE


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

065510  011646                $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
065512  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
065520  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
065522  065510                $TRPAD: .WORD   $TRAP2
065524  062654                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
065526  062452                        $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
065530  062426                        $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
065532  062466                        $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
065534  062202                        $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
065536  062126                        $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

065540  064476                        $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING

065542  064406                        $CKSWR  ;;CALL=CKSWR    TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
065544  064750                        $RDCHR  ;;CALL=RDCHR    TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
065546  065040                        $RDLIN  ;;CALL=RDLIN    TRAP+12(104412) TTY TYPEIN STRING ROUTINE
065550  065346                        $RDOCT  ;;CALL=RDOCT    TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
065552  062032                        $SAVREG ;;CALL=SAVREG   TRAP+14(104414) SAVE R0-R5 ROUTINE
065554  062070                        $RESREG ;;CALL=RESREG   TRAP+15(104415) RESTORE R0-R5 ROUTINE
11                              .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**************************************************************
                                ;POWER DOWN ROUTINE
065556  012737  065716  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
065564  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
065572  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
065574  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
065576  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
065600  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
065602  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
065604  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
065606  017746  113342                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
065612  010637  065722                MOV     SP,$SAVR6       ;;SAVE SP
065616  012737  065630  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
065624  000000                        HALT
065626  000776                        BR      .-2             ;;HANG UP
```

```
                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
065630  012737  065716  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
065636  013706  065722                   MOV     $SAVR6,SP        ;;GET SP
065642  005037  065722                   CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
065646  005237  065722           1$:     INC     $SAVR6           ;;WAIT FOR THE INC
065652  001375                           BNE     1$               ;;OF  WORD
065654  012677  113274                   MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
065660  012605                           MOV     (SP)+,R5         ;;POP STACK INTO R5
065662  012604                           MOV     (SP)+,R4         ;;POP STACK INTO R4
065664  012603                           MOV     (SP)+,R3         ;;POP STACK INTO R3
065666  012602                           MOV     (SP)+,R2         ;;POP STACK INTO R2
065670  012601                           MOV     (SP)+,R1         ;;POP STACK INTO R1
065672  012600                           MOV     (SP)+,R0         ;;POP STACK INTO R0
065674  012737  065556  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
065702  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
065710  104401                           TYPE                     ;REPORT THE POWER FAILURE
065712  065724                   $PWRMG: .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
065714  000002                           RTI
065716  000000                   $ILLUP: HALT                     ;;THE POWER UP SEQUENCE WAS STARTED
065720  000776                           BR      .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
065722  000000                   $SAVR6: 0                        ;;PUT THE SP HERE
065724     015     012     120   $POWER: .ASCIZ  <15><12>''POWER''
                                        .EVEN
  12                                    .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;*****************************************************************
065734  112737  000001  066200  $ATY1:  MOVB    #1,$FFLG         ;;TO REPORT FATAL ERROR
065742  112737  000001  066176  $ATY3:  MOVB    #1,$MFLG         ;;TO TYPE A MESSAGE
065750  000403                           BR      $ATYC
065752  112737  000001  066200  $ATY4:  MOVB    #1,$FFLG         ;;TO ONLY REPORT FATAL ERROR
065760                           $ATYC:
065760  010046                           MOV     R0,-(SP)         ;;PUSH R0 ON STACK
065762  010146                           MOV     R1,-(SP)         ;;PUSH R1 ON STACK
065764  105737  066176                   TSTB    $MFLG            ;;SHOULD TYPE A MESSAGE?
065770  001450                           BEQ     5$               ;;IF NOT:  BR
065772  122737  000001  001242          CMPB    #APTENV,$ENV     ;;OPERATING UNDER APT?
066000  001031                           BNE     3$               ;;IF NOT:  BR
066002  132737  000100  001243          BITB    #APTSPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
066010  001425                           BEQ     3$               ;;IF NOT:  BR
066012  017600  000004                   MOV     @4(SP),R0        ;;GET MESSAGE ADDR.
066016  062766  000002  000004          ADD     #2,4(SP)                 ;;BUMP RETURN ADDR.
066024  005737  001222           1$:     TST     $MSGTYPE         ;;SEE IF DONE W/ LAST XMISSION?
066030  001375                           BNE     1$               ;;IF NOT:  WAIT
066032  010037  001236                   MOV     R0,$MSGAD        ;;PUT ADDR IN MAILBOX
066036  105720                   2$:     TSTB    (R0)+            ;;FIND END OF MESSAGE
066040  001376                           BNE     2$
066042  163700  001236                   SUB     $MSGAD,R0        ;;SUB START OF MESSAGE
066046  006200                           ASR     R0               ;;GET MESSAGE LNGTH IN WORDS
066050  010037  001240                   MOV     R0,$MSGLGT       ;;PUT LENGTH IN MAILBOX
066054  012737  000004  001222          MOV     #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
066062  000413                           BR      5$
066064  017637  000004  066110  3$:     MOV     @4(SP),4$        ;;PUT MSG ADDR IN JSR LINKAGE
066072  062766  000002  000004          ADD     #2,4(SP)                 ;;BUMP RETURN ADDRESS
066100  013746  177776                   MOV     177776,-(SP)     ;;PUSH 177776 ON STACK
066104  004737  062654                   JSR     PC,$TYPE         ;;CALL TYPE MACRO
```

```
066110  000000                  4$:     .WORD   0
066112                          5$:
066112  105737  066200          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
066116  001416                          BEQ     12$             ;;IF NOT:  BR
066120  005737  001242                  TST     $ENV            ;;RUNNING UNDER APT?
066124  001413                          BEQ     12$             ;;IF NOT:  BR
066126  005737  001222          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
066132  001375                          BNE     11$             ;;IF NOT:  WAIT
066134  017637  000004  001224          MOV     @4(SP),$FATAL   ;;GET ERROR #
066142  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
066150  005237  001222                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
066154  105037  066200          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
066160  105037  066177                  CLRB    $LFLG           ;;CLEAR LOG FLAG
066164  105037  066176                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
066170  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
066172  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
066174  000207                          RTS     PC              ;;RETURN
066176     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
066177     000          $LFLG:  .BYTE   0               ;;LOG FLAG
066200     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
                                .EVEN

            000200      APTSIZE = 200
            000001      APTENV  = 001
            000100      APTSPOOL= 100
            000040      APTCSUP = 040
```

```
                                        .SBTTL  CONSOLE MESSAGES

066202  075     000             EQUALS: .ASCIZ  @=@
066204  101     114     114     ALL:    .ASCIZ  @ALL@<CRLF>
066211  040     077     040     QUES:   .ASCIZ  @ ? @
066215  054     040     000     COMMA:  .ASCIZ  @, @
066220  200     124     131     MSHELP: .ASCIZ  <CRLF>@TYPE HELP TEXT (Y/N) ? @
066251                          UBUSQST:
066251  200     103     110             .ASCIZ  <CRLF>@CHANGE ADDRESSES (Y/N) ? @
066304  200     125     123     CNSL00. .ASCIZ  <CRLF>@USE SAME DEVICES (Y/N) ? @
066337  200     102     125     CNSL01: .ASCIZ  <CRLF>@BUS ADDRESS @
066355  040     114     111     CNSL02: .ASCIZ  @ LIMITS - LO= 160000, HI= 17XXXX@<CRLF>
066417  126     105     103     CNSL03: .ASCIZ  @VECTOR ADDRESS @
066437  040     114     111     CNSL04: .ASCIZ  @ LIMITS - LO= 0, HI= 1000@<CRLF><LF>
066473  102     122     040     CNSL05: .ASCIZ  @BR LEVEL @
066505  040     114     111     CNSL06: .ASCIZ  @ LIMITS - LO= 0, HI= 7@<CRLF><LF>
066536  200                     CNSL07: .ASCII  <CRLF>
066537  200     124     131             .ASCII  <CRLF>@TYPE "A" TO TEST ALL DRIVES, OR TYPE DRIVE NUMBER(S)@
066624  200     101     116             .ASCIZ  <CRLF>@AND TERMINATE INPUT WITH A CARRIAGE RETURN.@
066701  200                     CNSL08: .ASCII  <CRLF>
066702  040     077     111     CNSL09: .ASCIZ  @ ?ILLEGAL INPUT@<CRLF>
066723  200     104     122     MSDRVS: .ASCIZ  <CRLF>/DRIVE(S): /
066737  104     122     111     MSGDRV: .ASCIZ  /DRIVE/
066745  040     111     123     LODEV:  .ASCIZ  / IS LOAD DEVICE/
066765  040     116     117     NOTPRS: .ASCIZ  / NOT PRESENT/
067002  040     116     117     NOTAVL: .ASCIZ  / NOT AVAILABLE/
067021  040     117     106     OFFLIN: .ASCIZ  / OFF LINE/
067033  040     117     116     ONLINE: .ASCIZ  / ON LINE /

                                        .EVEN
```

```
                                .SBTTL  FUNCTION CODE TABLE

                                ;THE FUNCTION CODE TABLE IS USED TO DEFINE STATUS CONDITIONS FOR
                                ;EACH FUNCTION CODE.  BIT USAGE IS AS FOLLOWS:

                                ;       ATA - BIT 15 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF ATA SHOULD BE SET WHEN THE FUNCTION CODE IS EXECUTED, OTHERWISE,
                                ;BIT 15 IS ZERO, INDICATING THAT ATA SHOULD NOT NORMALLY BE SET.
                                ;NOTE THAT ATA MAY BE SET WHEN A COMMAND IS EXECUTED EVEN THOUGH
                                ;IT IS NOT EXPECTED AS A RESULT OF THE COMMAND.

                                ;       WCE - BIT 14 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF WRITE CHECK ERRORS ARE ENABLED AS A FUNCTION OF THE COMMAND.

                                ;       OPI - BIT 13 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF OPI ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.

                                ;       IVC - BIT 12 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF IVC ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.

                                ;       WLE - BIT 11 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF WRITE ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.
                                ;THE WRITE ERRORS WHICH ARE ENABLED ARE "WLE", "WCF", "DPE", "UPE".

                                ;       IAE - BIT 10 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF INVALID ADDRESS ERROR IS ENABLED FOR THAT COMMAND.

                                ;       AOE - BIT 09 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF READ AND WRITE ERRORS ARE ENABLED DURING THE EXECUTION OF THE
                                ;COMMAND.  THE ERRORS ENABLED BY THIS BIT ARE "TRE", "DLT", "NEM",
                                ;"MXF", "LBT", AND "AOE".

                                ;               BIT 08 IS NOT USED.

                                ;       HCE - BIT 07 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF HEADER ERRORS ARE ENABLED DURING THE EXECUTION OF THAT COMMAND.
                                ;HEADER ERRORS INCLUDE "HCRC", "HCE", "FER" AND "BSE"".

                                ;       ECH - BIT 06 IS SET IN THE ENTRY FOR A GIVEN FUNCTION CODE
                                ;IF DATA FIELD ERRORS ARE ENABLED DURING THE EXECUTION OF THA"
                                ;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.

067046                          FNCDTB:                                 ;FUNCTION CODE TABLE

067046  020000                  .WORD   OPI                             ;NOP
```

```
     58 067050  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (2)
     59 067052  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEEK
     60 067054  130000                .WORD   ATA!OPI!IVC                     ;RECALIBRATE
     61 067056  020000                .WORD   OPI                             ;DRIVE CLEAR
     62 067060  030000                .WORD   OPI!IVC                         ;RELEASE
     63 067062  130000                .WORD   OPI!ATA!IVC                     ;OFFSET
     64 067064  130000                .WORD   OPI!ATA!IVC                     ;RETURN TO CENTERLINE
     65 067066  020000                .WORD   OPI                             ;READ IN PRESET
     66 067070  020000                .WORD   OPI                             ;PACK ACKNOWLEDGE
     67 067072  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (24)
     68 067074  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (26)
     69 067076  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEARCH
     70 067100  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (32)
     71 067102  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (34)
     72 067104  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (36)
     73 067106  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (40)
     74 067110  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (42)
     75 067112  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (44)
     76 067114  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (46)
     77 067116  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
     78 067120  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK HEADER AND DATA
     79 067122  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (54)
     80 067124  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (56)
     81 067126  037200                .WORD   OPI!IVC!WLE!IAE!AOE!HCE         ;WRITE DATA
     82 067130  037000                .WORD   OPI!IVC!WLE!IAE!AOE             ;WRITE HEADER AND DATA
     83 067132  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (64)
     84 067134  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (66)
     85 067136  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ DATA
     86 067140  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ HEADER AND DATA
     87 067142  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (74)
     88 067144  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (76)
     89
```

SEQ 0260

```
                                .SBTTL  ATTENTION (ATA) TABLE

                        ATNTBL: .BYTE   1.
                                .BYTE   2.
                                .BYTE   4.
                                .BYTE   8.
                                .BYTE   16.
                                .BYTE   32.
                                .BYTE   64.
                                .BYTE   128.
```

```
                                .SBTTL  DATA PATTERN TABLE

                        RGDTPT:
                        MIXED:
067156 000000                   .WORD   0.
067160 000001                   .WORD   1.
067162 000003                   .WORD   3.
067164 000007                   .WORD   7.
067166 000017                   .WORD   15.
067170 000037                   .WORD   31.
067172 000077                   .WORD   63.
067174 000177                   .WORD   127.
067176 000377                   .WORD   255.
067200 000777                   .WORD   511.
067202 001777                   .WORD   1023.
067204 003777                   .WORD   2047.
067206 007777                   .WORD   4095.
067210 017777                   .WORD   8191.
067212 037777                   .WORD   16383.
067214 077777                   .WORD   32767.
067216 177777           ONES:   .WORD   65535.
067220 177777                   .WORD   65535.
067222 077777                   .WORD   32767.
067224 037777                   .WORD   16383.
067226 017777                   .WORD   8191.
067230 007777                   .WORD   4095.
067232 003777                   .WORD   2047.
067234 001777                   .WORD   1023.
067236 000777                   .WORD   511.
067240 000377                   .WORD   255.
067242 000177                   .WORD   127.
067244 000077                   .WORD   63.
067246 000037                   .WORD   31.
067250 000017                   .WORD   15.
067252 000007                   .WORD   7.
067254 000003                   .WORD   3.
067256 000001                   .WORD   1.
067260 000000           ZEROS:  .WORD   0.
067262 000000                   .WORD   0.
067264 000001                   .WORD   1.
067266 000002                   .WORD   2.
067270 000004                   .WORD   4.
067272 000010                   .WORD   8.
067274 000020                   .WORD   16.
067276 000040                   .WORD   32.
067300 000100                   .WORD   64.
067302 000200                   .WORD   128.
067304 000400                   .WORD   256.
067306 001000                   .WORD   512.
067310 002000                   .WORD   1024.
067312 004000                   .WORD   2048.
067314 010000                   .WORD   4096.
067316 020000                   .WORD   8192.
067320 040000                   .WORD   16384.
067322 100000                   .WORD   32768.
067324 100000                   .WORD   32768.
067326 040000                   .WORD   16384.
```

```
 58 067330  020000                    .WORD   8192.
 59 067332  010000                    .WORD   4096.
 60 067334  004000                    .WORD   2048.
 61 067336  002000                    .WORD   1024.
 62 067340  001000                    .WORD   512.
 63 067342  000400                    .WORD   256.
 64 067344  000200                    .WORD   128.
 65 067346  000100                    .WORD   64.
 66 067350  000040                    .WORD   32.
 67 067352  000020                    .WORD   16.
 68 067354  000010                    .WORD   8.
 69 067356  000004                    .WORD   4.
 70 067360  000002                    .WORD   2.
 71 067362  000001                    .WORD   1.
 72 067364  000000                    .WORD   0.
 73 067366  177777                    .WORD   65535.
 74 067370  177776                    .WORD   65534.
 75 067372  177774                    .WORD   65532.
 76 067374  177770                    .WORD   65528.
 77 067376  177760                    .WORD   65520.
 78 067400  177740                    .WORD   65504.
 79 067402  177700                    .WORD   65472.
 80 067404  177600                    .WORD   65408.
 81 067406  177400                    .WORD   65280.
 82 067410  177000                    .WORD   65024.
 83 067412  176000                    .WORD   64512.
 84 067414  174000                    .WORD   63488.
 85 067416  170000                    .WORD   61440.
 86 067420  160000                    .WORD   57344.
 87 067422  140000                    .WORD   49152.
 88 067424  100000                    .WORD   32768.
 89 067426  000000                    .WORD   0.
 90 067430  000000                    .WORD   0.
 91 067432  100000                    .WORD   32768.
 92 067434  140000                    .WORD   49152.
 93 067436  160000                    .WORD   57344.
 94 067440  170000                    .WORD   61440.
 95 067442  174000                    .WORD   63488.
 96 067444  176000                    .WORD   64512.
 97 067446  177000                    .WORD   65024.
 98 067450  177400                    .WORD   65280.
 99 067452  177600                    .WORD   65408.
100 067454  177700                    .WORD   65472.
101 067456  177740                    .WORD   65504.
102 067460  177760                    .WORD   65520.
103 067462  177770                    .WORD   65528.
104 067464  177774                    .WORD   65532.
105 067466  177776                    .WORD   65534.
106 067470  177777                    .WORD   65535.
107 067472  125252          EARLY:    .WORD   43690.
108 067474  152525                    .WORD   43690./2
109 067476  125252                    .WORD   43690.
110 067500  177777                    .WORD   65535.
111 067502  177776                    .WORD   65534.
112 067504  177775                    .WORD   65533.
113 067506  177773                    .WORD   65531.
114 067510  177767                    .WORD   65527.
```

```
115 067512  177757                    .WORD   65519.
116 067514  177737                    .WORD   65503.
117 067516  177677                    .WORD   65471.
118 067520  177577                    .WORD   65407.
119 067522  177377                    .WORD   65279.
120 067524  176777                    .WORD   65023.
121 067526  175777                    .WORD   64511.
122 067530  173777                    .WORD   63487.
123 067532  167777                    .WORD   61439.
124 067534  157777                    .WORD   57343.
125 067536  137777                    .WORD   49151.
126 067540  077777                    .WORD   32767.
127 067542  077777                    .WORD   32767.
128 067544  137777                    .WORD   49151.
129 067546  157777                    .WORD   57343.
130 067550  167777                    .WORD   61439.
131 067552  173777                    .WORD   63487.
132 067554  175777                    .WORD   64511.
133 067556  176777                    .WORD   65023.
134 067560  177377                    .WORD   65279.
135 067562  177577                    .WORD   65407.
136 067564  177677                    .WORD   65471.
137 067566  177737                    .WORD   65503.
138 067570  177757                    .WORD   65519.
139 067572  177767                    .WORD   65527.
140 067574  177773                    .WORD   65531.
141 067576  177775                    .WORD   65533.
142 067600  177776                    .WORD   65534.
143 067602  177777                    .WORD   65535.
144 067604                            ENRGDT:
145
```

```
                                .SBTTL  ERROR MESSAGE TABLE

067604  074200  000000          EMT1:   .WORD   EMS1,0
067610  074247  074264  000000  EMT2:   .WORD   EMS2,EMS3,0
067616  074247  074327  000000  EMT3:   .WORD   EMS2,EMS4,0
067624  074372  074422  000000  EMT4:   .WORD   EMS5,EMS6,0
067632  074372  074534  000000  EMT5:   .WORD   EMS5,EMS10,0
067640  101175  076413  000000  EMT6:   .WORD   EMS167,EMS64,0
067646  077161  101222  000000  EMT7:   .WORD   EMS110,EMS170,0
067654  074467  000000          EMT10:  .WORD   EMS7,0
067660  074534  000000          EMT11:  .WORD   EMS10,0
067664  074576  074607  000000  EMT12:  .WORD   EMS11,EMS12,0
067672  074650  074661  074672  EMT13:  .WORD   EMS13,EMS14,EMS15,EMS16,0
067704  074744  076413  000000  EMT14:  .WORD   EMS17,EMS64,0
067712  074576  075027  000000  EMT15:  .WORD   EMS11,EMS21,0
067720  074576  075052  075201  EMT16:  .WORD   EMS11,EMS22,EMS27,0
067730  074576  075066  075212  EMT17:  .WORD   EMS11,EMS23,EMS30,0
067740  074576  075114  075212  EMT20:  .WORD   EMS11,EMS24,EMS30,0
067750  074576  075143  075201  EMT21:  .WORD   EMS11,EMS25,EMS27,0
067760  074576  075160  075201  EMT22:  .WORD   EMS11,EMS26,EMS27,0
067770  074576  075222  075212  EMT23:  .WORD   EMS11,EMS31,EMS30,0
070000  074576  075251  075212  EMT24:  .WORD   EMS11,EMS32,EMS30,0
070010  074576  075300  075212  EMT25:  .WORD   EMS11,EMS33,EMS30,0
070020  074576  075326  075212  EMT26:  .WORD   EMS11,EMS34,EMS30,0
070030  074576  075377  075212  EMT27:  .WORD   EMS11,EMS35,EMS30,0
070040  074576  075426  075212  EMT30:  .WORD   EMS11,EMS36,EMS30,0
070050  074576  075455  075212  EMT31:  .WORD   EMS11,EMS37,EMS30,0
070060  074576  075503  075212  EMT32:  .WORD   EMS11,EMS40,EMS30,0
070070  074576  075532  075212  EMT33:  .WORD   EMS11,EMS41,EMS30,0
070100  074576  075560  075212  EMT34:  .WORD   EMS11,EMS42,EMS30,0
070110  074576  075607  075212  EMT35:  .WORD   EMS11,EMS43,EMS30,0
070120  074576  075636  075212  EMT36:  .WORD   EMS11,EMS44,EMS30,0
070130  074576  075711  075212  EMT37:  .WORD   EMS11,EMS45,EMS30,0
070140  076477  075004  000000  EMT40:  .WORD   EMS66,EMS20,0
070146  076665  100376  076673  EMT41:  .WORD   EMS75,EMS141,EMS76,0
070156  100467  100477  076621  EMT42:  .WORD   EMS144,EMS145,EMS72,EMS76,0
070170  076003  076117  076673  EMT43:  .WORD   EMS47,EMS53,EMS76,0
070200  076724  076117  076673  EMT44:  .WORD   EMS77,EMS53,EMS76,0
070210  076752  076117  076673  EMT45:  .WORD   EMS100,EMS53,EMS76,0
070220  077000  076117  076673  EMT46:  .WORD   EMS101,EMS53,EMS76,0
070230  076431  076413  000000  EMT47:  .WORD   EMS65,EMS64,0
070236  074650  074672  076365  EMT50:  .WORD   EMS13,EMS15,EMS63,0
070246  074576  075754  075212  EMT51:  .WORD   EMS11,EMS46,EMS30,EMS67,0
070260  076003  076117  076547  EMT52:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS140,EMS141,0
070276  076003  076117  076547  EMT53:  .WORD   EMS47,EMS53,EMS67,EMS115,EMS141,EMS164,0
070314  076032  076117  076547  EMT54:  .WORD   EMS50,EMS53,EMS67,0
070324  100424  100441  076117  EMT55:  .WORD   EMS142,EMS143,EMS53,EMS67,0
070336  076061  076621  076547  EMT56:  .WORD   EMS51,EMS72,EMS67,EMS115,EMS50,EMS70,0
070354  101175  076631  076547  EMT57:  .WORD   EMS167,EMS73,EMS67,0
070364  076202  076547  077361  EMT60:  .WORD   EMS56,EMS67,EMS115,EMS150,EMS152,EMS70,0
070402  076606  076202  076547  EMT61:  .WORD   EMS71,EMS56,EMS67,EMS115,EMS150,EMS152,EMS72,0
070422  100355  076547  077361  EMT62:  .WORD   EMS140,EMS67,EMS115,EMS47,EMS70,0
070436  000000                  EMT63:  .WORD
070440  100562  100625  076574  EMT64:  .WORD   EMS150,EMS152,EMS70,EMS115,EMS56,EMS73,EMS67,0
070460  076107  101622  102003  EMT65:  .WORD   EMS52,EMS205,EMS214,EMS206,EMS115,EMS51,EMS72,0
070500  101134  077055  076621  EMT66:  .WORD   EMS165,EMS103,EMS72,EMS124,0
070512  101134  077055  076621  EMT67:  .WORD   EMS165,EMS103,EMS72,EMS171,0
```

```
 58 070524  075754  075004  101027  EMT70:  .WORD   EMS46,EMS20,EMS163,0
 59 070534  076606  077000  101027  EMT71:  .WORD   EMS71,EMS101,EMS163,0
 60 070544  076003  076621  101027  EMT72:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS140,EMS141,0
 61 070562  076003  076621  101027  EMT73:  .WORD   EMS47,EMS72,EMS163,EMS115,EMS141,EMS72,0
 62 070600  076202  076117  101027  EMT74:  .WORD   EMS56,EMS53,EMS163,0
 63 070610  076606  076202  101027  EMT75:  .WORD   EMS71,EMS56,EMS163,EMS115,EMS150,EMS152,EMS72,0
 64 070630  076032  076117  101027  EMT76:  .WORD   EMS50,EMS53,EMS163,0
 65 070640  100424  100441  076117  EMT77:  .WORD   EMS142,EMS143,EMS53,EMS163,0
 66 070652  076665  100376  101027  EMT100: .WORD   EMS75,EMS141,EMS163,EMS115,EMS47,EMS70,0
 67 070670  076665  100562  101027  EMT101: .WORD   EMS75,EMS150,EMS163,EMS115,EMS56,EMS73,0
 68 070706  101175  076631  101027  EMT102: .WORD   EMS167,EMS73,EMS163,0
 69 070716  100545  100601  076646  EMT103: .WORD   EMS147,EMS151,EMS74,EMS163,0
 70 070730  076061  076646  101027  EMT104: .WORD   EMS51,EMS74,EMS163,EMS115,EMS50,EMS70,0
 71 070746  101134  076752  101027  EMT105: .WORD   EMS165,EMS100,EMS163,0
 72 070756  101134  076724  101027  EMT106: .WORD   EMS165,EMS77,EMS163,0
 73 070766  100467  100477  076117  EMT107: .WORD   EMS144,EMS145,EMS53,EMS163,EMS115,EMS143,EMS70,0
 74 071006  077161  077221  077364  EMT110: .WORD   EMS110,EMS112,EMS116,EMS111,0
 75 071020  077305  074327  000000  EMT111: .WORD   EMS113,EMS4,0
 76 071026  077305  074264  000000  EMT112: .WORD   EMS113,EMS3,0
 77 071034  077161  077361  077364  EMT113: .WORD   EMS110,EMS115,EMS116,EMS117,EMS114,0
 78 071050  077305  077436  000000  EMT114: .WORD   EMS113,EMS120,0
 79 071056  077457  100117  100143  EMT115: .WORD   EMS121,EMS132,EMS133,0
 80 071066  077522  100117  100143  EMT116: .WORD   EMS122,EMS132,EMS133,0
 81 071076  077557  100117  100143  EMT117: .WORD   EMS123,EMS132,EMS133,0
 82 071106  077622  100117  100143  EMT120: .WORD   EMS124,EMS132,EMS133,
 83 071116  077654  100117  100143  EMT121: .WORD   EMS125,EMS132,EMS133,0
 84 071126  077717  100117  100143  EMT122: .WORD   EMS126,EMS132,EMS133,0
 85 071136  100320  100117  100143  EMT123: .WORD   EMS137,EMS132,EMS133,0
 86 071146  100021  100117  100143  EMT124: .WORD   EMS130,EMS132,EMS133,0
 87 071156  100057  100117  100143  EMT125: .WORD   EMS131,EMS132,EMS133,0
 88 071166  077457  100117  100166  EMT126: .WORD   EMS121,EMS132,EMS134,EMS123,0
 89 071200  077522  100117  100166  EMT127: .WORD   EMS122,EMS132,EMS134,EMS123,0
 90 071212  077557  100117  100166  EMT130: .WORD   EMS123,EMS132,EMS134,EMS123,0
 91 071224  077622  100117  100166  EMT131: .WORD   EMS124,EMS132,EMS134,EMS123,0
 92 071236  077654  100117  100166  EMT132: .WORD   EMS125,EMS132,EMS134,EMS123,0
 93 071250  077717  100117  100166  EMT133: .WORD   EMS126,EMS132,EMS134,EMS123,0
 94 071262  100320  100117  100166  EMT134: .WORD   EMS137,EMS132,EMS134,EMS123,0
 95 071274  100021  100117  100166  EMT135: .WORD   EMS130,EMS132,EMS134,EMS123,0
 96 071306  100057  100117  100166  EMT136: .WORD   EMS131,EMS132,EMS134,EMS123,0
 97 071320  077457  100117  100230  EMT137: .WORD   EMS121,EMS132,EMS135,0
 98 071330  077557  100117  100230  EMT140: .WORD   EMS123,EMS132,EMS135,0
 99 071340  077457  100117  100273  EMT141: .WORD   EMS121,EMS132,EMS136,0
100 071350  100320  100117  100273  EMT142: .WORD   EMS137,EMS132,EMS136,0
101 071360  077622  100117  100273  EMT143: .WORD   EMS124,EMS132,EMS136,0
102 071370  077654  100117  100273  EMT144: .WORD   EMS125,EMS132,EMS136,0
103 071400  077717  100117  100273  EMT145: .WORD   EMS126,EMS132,EMS136,0
104 071410  100057  100117  100273  EMT146: .WORD   EMS131,EMS132,EMS136,0
105 071420  101251  100117  100273  EMT147: .WORD   EMS171,EMS132,EMS136,0
106 071430  100021  100117  100273  EMT150: .WORD   EMS130,EMS132,EMS136,0
107 071440  100355  077361  100376  EMT151: .WORD   EMS140,EMS115,EMS141,EMS70,0
108 071452  100424  077361  100441  EMT152: .WORD   EMS142,EMS115,EMS143,EMS72,0
109 071464  100467  100477  100526  EMT153: .WORD   EMS144,EMS145,EMS146,EMS115,EMS143,EMS72,0
110 071502  100467  100477  075004  EMT154: .WORD   EMS144,EMS145,EMS20,EMS115,EMS143,EMS70,0
111 071520  100562  100625  100673  EMT155: .WORD   EMS150,EMS152,EMS154,EMS153,0
112 071532  100545  100601  100673  EMT156: .WORD   EMS147,EMS151,EMS154,EMS155,0
113 071544  100545  100601  100727  EMT157: .WORD   EMS147,EMS151,EMS156,EMS157,0
114 071556  100777  100727  100762  EMT160: .WORD   EMS161,EMS156,EMS160,0
```

```
115 071566  075143  075201  100727  EMT161: .WORD   EMS25,EMS27,EMS156,EMS160,0
116 071600  075160  075201  100727  EMT162: .WORD   EMS26,EMS27,EMS156,EMS160,0
117 071612  076003  101027  100355  EMT163: .WORD   EMS47,EMS163,EMS140,0
118 071622  076003  075004  000000  EMT164: .WORD   EMS47,EMS20,0
119 071630  076202  075004  000000  EMT165: .WORD   EMS56,EMS20,0
120 071636  075754  075004  000000  EMT166: .WORD   EMS46,EMS20,0
121 071644  102323  075004  000000  EMT167: .WORD   EMS224,EMS20,0
122 071652  101175  100673  101150  EMT170: .WORD   EMS167,EMS154,EMS166,0
123 071662  101175  100673  100745  EMT171: .WORD   EMS167,EMS154,EMS157,0
124 071672  101175  100673  100707  EMT172: .WORD   EMS167,EMS154,EMS155,0
125 071702  101251  100117  100143  EMT173: .WORD   EMS171,EMS132,EMS133,0
126 071712  101251  100117  100166  EMT174: .WORD   EMS171,EMS132,EMS134,EMS123,0
127 071724  077305  101424  000000  EMT175: .WORD   EMS113,EMS177,0
128 071732  101446  101463  000000  EMT176: .WORD   EMS200,EMS201,0
129 071740  101561  100376  075212  EMT177: .WORD   EMS203,EMS141,EMS30,EMS202,0
130 071752  101561  076061  075212  EMT200: .WORD   EMS203,EMS51,EMS30,EMS202,0
131 071764  101561  101574  075212  EMT201: .WORD   EMS203,EMS204,EMS30,EMS202,0
132 071776  101561  076032  075212  EMT202: .WORD   EMS203,EMS50,EMS30,EMS202,0
133 072010  101561  100441  075212  EMT203: .WORD   EMS203,EMS143,EMS30,EMS202,0
134 072022  100562  076646  101531  EMT204: .WORD   EMS150,EMS74,EMS202,EMS115,EMS152,EMS72,0
135 072040  076032  077055  076621  EMT205: .WORD   EMS50,EMS103,EMS72,0
136 072050  077064  076631  101531  EMT206: .WORD   EMS104,EMS73,EMS202,0
137 072060  076665  100376  077361  EMT207: .WORD   EMS75,EMS141,EMS115,EMS140,0
138 072072  076665  100562  000000  EMT210: .WORD   EMS75,EMS150,0
139 072100  076061  076621  077361  EMT211: .WORD   EMS51,EMS72,EMS115,EMS50,EMS70,0
140 072114  100424  076117  077361  EMT212: .WORD   EMS142,EMS53,EMS115,EMS143,EMS72,0
141 072130  076032  077055  076117  EMT213: .WORD   EMS50,EMS103,EMS53,0
142 072140  076107  101622  101150  EMT214: .WORD   EMS52,EMS205,EMS166,EMS206,EMS115,EMS51,EMS72,0
143 072160  076107  077414  076202  EMT215: .WORD   EMS52,EMS117,EMS56,EMS163,0
144 072172  076202  075212  076413  EMT216: .WORD   EMS56,EMS30,EMS64,0
145 072202  075754  075212  076413  EMT217: .WORD   EMS46,EMS30,EMS64,0
146 072212  075222  075212  076413  EMT220: .WORD   EMS31,EMS30,EMS64,0
147 072222  076003  075212  076413  EMT221: .WORD   EMS47,EMS30,EMS64,0
148 072232  076107  077414  076724  EMT222: .WORD   EMS52,EMS117,EMS77,0
149 072242  076107  077414  076365  EMT223: .WORD   EMS52,EMS117,EMS63,0
150 072252  000000                  EMT224: .WORD
151 072254  000000                  EMT225: .WORD
152 072256  000000                  EMT226: .WORD
153 072260  000000                  EMT227: .WORD
154 072262  000000                  EMT230: .WORD
155 072264  000000                  EMT231: .WORD
156 072266  000000                  EMT232: .WORD
157 072270  000000                  EMT233: .WORD
158 072272  000000                  EMT234: .WORD
159 072274  000000                  EMT235: .WORD
160 072276  000000                  EMT236: .WORD
161 072300  000000                  EMT237: .WORD
162 072302  000000                  EMT240: .WORD
163 072304  000000                  EMT241: .WORD
164 072306  000000                  EMT242: .WORD
165 072310  000000                  EMT243: .WORD
166 072312  000000                  EMT244: .WORD
167 072314  000000                  EMT245: .WORD
168 072316  101175  100117  101661  EMT246: .WORD   EMS167,EMS132,EMS207,0
169 072326  101175  100117  101706  EMT247: .WORD   EMS167,EMS132,EMS210,EMS125,0
170 072340  101175  101717  101706  EMT250: .WORD   EMS167,EMS211,EMS210,EMS207,EMS206,0
171 072354  101735  101760  000000  EMT251: .WORD   EMS212,EMS213,0
```

```
172 072362  101134  101735  000000  EMT252: .WORD   EMS165,EMS212,0
173 072370  100545  100601  100727  EMT253: .WORD   EMS147,EMS151,EMS156,EMS210,EMS26,EMS27,0
174 072406  101561  077000  075212  EMT254: .WORD   EMS203,EMS101,EMS30,0
175 072416  101561  101175  075212  EMT255: .WORD   EMS203,EMS167,EMS30,0
176 072426  101561  076752  075212  EMT256: .WORD   EMS203,EMS100,EMS30,0
177 072436  074576  075754  075212  EMT257: .WORD   EMS11,EMS46,EMS30,EMS102,0
178 072450  076107  077414  076202  EMT260: .WORD   EMS52,EMS117,EMS56,EMS102,0
179 072462  076107  101622  102125  EMT261: .WORD   EMS52,EMS205,EMS220,EMS206,EMS115,EMS51,EMS72,0
180 072502  077064  076631  101531  EMT262: .WORD   EMS104,EMS73,EMS202,0
181 072512  076032  076117  077026  EMT263: .WORD   EMS50,EMS53,EMS102,0
182 072522  076202  076117  077026  EMT264: .WORD   EMS56,EMS53,EMS102,EMS115,EMS150,EMS152,EMS70,0
183 072542  076606  076202  077026  EMT265: .WORD   EMS71,EMS56,EMS102,EMS115,EMS150,EMS152,EMS72,0
184 072562  100424  100441  076117  EMT266: .WORD   EMS142,EMS143,EMS53,EMS102,0
185 072574  076032  100526  077361  EMT267: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,0
186 072612  076003  076117  077026  EMT270: .WORD   EMS47,EMS53,EMS102,EMS115,EMS140,0
187 072626  076003  076117  077026  EMT271: .WORD   EMS47,EMS53,EMS102,EMS115,EMS141,EMS72,0
188 072644  076665  100376  077026  EMT272: .WORD   EMS75,EMS141,EMS102,EMS115,EMS47,EMS73,0
189 072662  076061  076646  077026  EMT273: .WORD   EMS51,EMS74,EMS102,EMS115,EMS50,EMS70,0
190 072700  076365  076117  076257  EMT274: .WORD   EMS63,EMS53,EMS57,EMS115,EMS41,EMS146,0
191 072716  077364  076117  075532  EMT275: .WORD   EMS116,EMS53,EMS41,EMS57,0
192 072730  076003  076117  076257  EMT276: .WORD   EMS47,EMS53,EMS57,EMS115,EMS140,0
193 072744  076003  076117  076257  EMT277: .WORD   EMS47,EMS53,EMS57,EMS115,EMS141,EMS72,0
194 072762  076202  076117  076257  EMT300: .WORD   EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS70,0
195 073002  076606  076202  076117  EMT301: .WORD   EMS71,EMS56,EMS53,EMS57,EMS115,EMS150,EMS152,EMS72,0
196 073024  101134  076752  077055  EMT302: .WORD   EMS165,EMS100,EMS103,EMS57,0
197 073036  101134  077000  077055  EMT303: .WORD   EMS165,EMS101,EMS103,EMS57,0
198 073050  101134  076724  077055  EMT304: .WORD   EMS165,EMS77,EMS103,EMS57,0
199 073062  075754  075212  076413  EMT305: .WORD   EMS46,EMS30,EMS64,EMS57,0
200 073074  076032  076117  076257  EMT306: .WORD   EMS50,EMS53,EMS57,0
201 073104  076032  100526  077361  EMT307: .WORD   EMS50,EMS146,EMS115,EMS52,EMS117,EMS46,EMS57,0
202 073124  100424  100441  076117  EMT310: .WORD   EMS142,EMS143,EMS53,EMS57,0
203 073136  077064  077055  076117  EMT311: .WORD   EMS104,EMS103,EMS53,EMS57,0
204 073150  077113  077055  076117  EMT312: .WORD   EMS105,EMS103,EMS53,EMS57,0
205 073162  100467  100477  077055  EMT313: .WORD   EMS144,EMS145,EMS103,EMS57,EMS115,EMS143,EMS70,0
206 073202  075560  077055  076117  EMT314: .WORD   EMS42,EMS103,EMS53,EMS57,0
207 073214  075222  077055  076117  EMT315: .WORD   EMS31,EMS103,EMS53,EMS57,0
208 073226  076606  075222  077055  EMT316: .WORD   EMS71,EMS31,EMS103,EMS57,0
209 073240  075607  077055  076257  EMT317: .WORD   EMS43,EMS103,EMS57,0
210 073250  075711  077055  076257  EMT320: .WORD   EMS45,EMS103,EMS57,0
211 073260  075636  077055  076257  EMT321: .WORD   EMS44,EMS103,EMS57,0
212 073270  077142  075004  000000  EMT322: .WORD   EMS106,EMS20,0
213 073276  075426  077055  076257  EMT323: .WORD   EMS36,EMS103,EMS57,0
214 073306  101324  075426  077055  EMT324: .WORD   EMS173,EMS36,EMS103,EMS57,0
215 073320  101304  075426  077055  EMT325: .WORD   EMS172,EMS36,EMS103,EMS57,0
216 073332  074650  101341  074672  EMT326: .WORD   EMS13,EMS174,EMS15,EMS35,EMS53,EMS175,0
217 073350  100545  100601  076646  EMT327: .WORD   EMS147,EMS151,EMS74,EMS175,0
218 073362  076477  076117  101347  EMT330: .WORD   EMS66,EMS53,EMS175,0
219 073372  075300  077055  076117  EMT331: .WORD   EMS33,EMS103,EMS53,EMS175,0
220 073404  075503  077055  076117  EMT332: .WORD   EMS40,EMS103,EMS53,EMS57,0
221 073416  076061  076646  076257  EMT333: .WORD   EMS51,EMS74,EMS57,EMS115,EMS50,EMS70,0
222 073434  076665  100376  076257  EMT334: .WORD   EMS75,EMS141,EMS57,EMS115,EMS47,EMS73,0
223 073452  076665  100562  100625  EMT335: .WORD   EMS75,EMS150,EMS152,EMS57,EMS115,EMS56,EMS73,0
224 073472  076305  076320  076347  EMT336: .WORD   EMS60,EMS61,EMS62,0
225 073502  077364  077414  075326  EMT337: .WORD   EMS116,EMS117,EMS34,0
226 073512  075326  076117  076131  EMT340: .WORD   EMS34,EMS53,EMS54,EMS111,0
227 073524  076153  075326  000000  EMT341: .WORD   EMS55,EMS34,0
228 073532  076107  077414  076202  EMT342: .WORD   EMS52,EMS117,EMS56,EMS57,0
```

```
    229 073544  076107  077414  075711  EMT343: .WORD   EMS52,EMS117,EMS45,EMS57,0
    230 073556  076107  077414  075636  EMT344: .WORD   EMS52,EMS117,EMS44,EMS57,0
    231 073570  076107  077414  102145  EMT345: .WORD   EMS52,EMS117,EMS221,0
    232 073600  077142  075004  077361  EMT346: .WORD   EMS106,EMS20,EMS115,EMS223,EMS72,0
    233 073614  076107  101622  102215  EMT347: .WORD   EMS52,EMS205,EMS222,EMS206,0
    234 073626  076665  100562  077026  EMT350: .WORD   EMS75,EMS150,EMS102,EMS115,EMS56,EMS73,0
    235 073644  101175  076631  077026  EMT351: .WORD   EMS167,EMS73,EMS102,0
    236 073654  102021  000000          EMT352: .WORD   EMS215,0
    237 073660  102072  101561  102042  EMT353: .WORD   EMS217,EMS203,EMS216,0
    238 073670  102145  075004  000000  EMT354: .WORD   EMS221,EMS20,0
    239
```

```
 1 073676  102375  103173  103250  EHT1:   .WORD   EH1,STSH1,STSH2,STSH4,0
 2 073710  103173  103250  103375  EHT2:   .WORD   STSH1,STSH2,STSH4,0
 3
 4 073720  102414  000000          EHT110: .WORD   EH110,0
 5 073724  102423  000000          EHT111: .WORD   EH111,0
 6
 7 073730  102442  000000          EHT114: .WORD   EH114,0
 8 073734  102471  103173  103250  EHT223: .WORD   EH223,STSH1,STSH2,STSH4,0
 9 073746  102517  103173  103250  EHT256: .WORD   EH256,STSH1,STSH2,STSH4,0
10
11 073760  102573  103173  103250  EHT336: .WORD   EH336,STSH1,STSH2,STSH4,0
12 073772  102632  103173  103250  EHT337: .WORD   EH337,STSH1,STSH2,STSH4,0
13 074004  102767  103173  103250  EHT344: .WORD   EH344,STSH1,STSH2,STSH4,0
14
15 074016  103125  000000          EHT353: .WORD   EH353,0
16
```

```
      1 074022  103434  103530  103546  EDT1:    .WORD   ED1,STSD1,STSD2,STSD4
      2 074032  103530  103546  103600  EDT2:    .WORD   STSD1,STSD2,STSD4
      3
      4 074040  103442                  EDT110:  .WORD   ED110
      5 074042  103446                  EDT111:  .WORD   ED111
      6
      7 074044  103454                  EDT114:  .WORD   ED114
      8 074046  103464  103530  103546  EDT223:  .WORD   ED223,STSD1,STSD2,STSD4
      9
     10 074056  103474  103530  103546  EDT336:  .WORD   ED336,STSD1,STSD2,STSD4
     11 074066  103506  103530  103546  EDT337:  .WORD   ED337,STSD1,STSD2,STSD4
     12 074076  103506  103530  103546  EDT344:  .WORD   ED337,STSD1,STSD2,STSD4,0
     13
     14 074110  103520                  EDT353:  .WORD   ED353
     15
```

```
      1 074112  103613  103631  103631  EFT1:    .WORD   EF111,STSF,STSF,STSF
      2 074122  103631  103631  103631  EFT2:    .WORD   STSF,STSF,STSF
      3
      4 074130  103612                  EFT110:  .WORD   EF110
      5 074132  103613                  EFT111:  .WORD   EF111
      6
      7 074134  103615                  EFT114:  .WORD   EF114
      8 074136  103615  103631  103631  EFT223:  .WORD   EF114,STSF,STSF,STSF
      9
     10 074146  103620  103631  103631  EFT336:  .WORD   EF336,STSF,STSF,STSF
     11 074156  103620  103631  103631  EFT337:  .WORD   EF336,STSF,STSF,STSF
     12 074166  103620  103631  103631  EFT344:  .WORD   EF336,STSF,STSF,STSF
     13
     14 074176  103615                  EFT353:  .WORD   EF114
     15
```

```
                                .SBTTL  ERROR MESSAGE STRINGS

074200  127  122  117  EMS1:   .ASCIZ  @WRONG UNIT SELECTED (RMCS2, BITS 0-2) @
074247  104  105  126  EMS2:   .ASCIZ  @DEVICE WENT @
074264  125  116  101  EMS3:   .ASCIZ  @UNAVAILABLE "DVA" (RMCS1, BIT 11) @
074327  116  117  116  EMS4:   .ASCIZ  @NONEXISTENT "NED" (RMCS2, BIT 12) @
074372  103  117  115  EMS5:   .ASCIZ  @COMMAND NOT COMPLETED, @
074422  103  117  116  EMS6:   .ASCIZ  @CONTROLLER NOT READY (RMCS1, BIT 7) @
074467  104  122  111  EMS7:   .ASCIZ  @DRIVE NOT READY "DRY" (RMDS, BIT 7) @
074534  107  117  040  EMS10:  .ASCIZ  @GO NOT RESET "GO" (RMCS1, BIT 0) @
074576  111  116  126  EMS11:  .ASCIZ  @INVALID @
074607  106  125  116  EMS12:  .ASCIZ  @FUNCTION CODE (RMCS1, BITS 1-5) @
074650  115  101  123  EMS13:  .ASCIZ  @MASSBUS @
074661  103  117  116  EMS14:  .ASCIZ  @CONTROL @
074672  102  125  123  EMS15:  .ASCIZ  @BUS PARITY ERROR @
074714  042  115  103  EMS16:  .ASCIZ  @"MCPE" (RMCS1, BIT 13) @
074744  124  122  101  EMS17:  .ASCIZ  @TRANSFER ERROR (RMCS1, BIT 14) @
075004  123  110  117  EMS20:  .ASCIZ  @SHOULD NOT BE SET @
075027  127  117  122  EMS21:  .ASCIZ  @WORD COUNT (RMWC) @
075052  102  125  123  EMS22:  .ASCIZ  @BUS (RMBA) @
075066  042  114  102  EMS23:  .ASCIZ  @"LBT" (RMDS, BIT 10) @
075114  042  101  117  EMS24:  .ASCIZ  @"AOE" (RMER1, BIT 09) @
075143  104  111  123  EMS25:  .ASCIZ  @DISK (RMDA) @
075160  103  131  114  EMS26:  .ASCIZ  @CYLINDER (RMDC) @
075201  101  104  104  EMS27:  .ASCIZ  @ADDRESS @
075212  123  124  101  EMS30:  .ASCIZ  @STATUS @
075222  042  127  114  EMS31:  .ASCIZ  @"WLE" (RMER1, BIT 11) @
075251  042  125  120  EMS32:  .ASCIZ  @"UPE" (RMCS2, BIT 13) @
075300  042  127  103  EMS33:  .ASCIZ  @"WCF" (RMER1, BIT 5) @
075326  127  122  111  EMS34:  .ASCIZ  @WRITE CHECK ERROR-"WCE" (RMCS2, BIT 14) @
075377  042  115  104  EMS35:  .ASCIZ  @"MDPE" (RMCS2, BIT 8) @
075426  042  104  103  EMS36:  .ASCIZ  @"DCK" (RMER1, BIT 15) @
075455  042  105  103  EMS37:  .ASCIZ  @"ECH" (RMER1, BIT 6) @
075503  042  104  114  EMS40:  .ASCIZ  @"DLT" (RMCS2, BIT 15) @
075532  042  115  130  EMS41:  .ASCIZ  @"MXF" (RMCS2, BIT 9) @
075560  042  104  124  EMS42:  .ASCIZ  @"DTE" (RMER1, BIT 12) @
075607  042  110  103  EMS43:  .ASCIZ  @"HCRC" (RMER1, BIT 8) @
075636  110  105  101  EMS44:  .ASCIZ  @HEADER COMPARE ERROR "HCE" (RMER1, BIT 7) @
075711  106  117  122  EMS45:  .ASCIZ  @FORMAT ERROR "FER" (RMER1, BIT 4) @
075754  042  111  101  EMS46:  .ASCIZ  @"IAE" (RMER1, BIT 10) @
076003  042  117  120  EMS47:  .ASCIZ  @"OPI" (RMER1, BIT 13) @
076032  042  123  113  EMS50:  .ASCIZ  @"SKI" (RMER2, BIT 14) @
076061  042  120  111  EMS51:  .ASCIZ  @"PIP" (RMDS, BIT 13) @
076107  124  110  105  EMS52:  .ASCIZ  @THE RM @
076117  104  105  124  EMS53:  .ASCIZ  @DETECTED @
076131  101  124  040  EMS54:  .ASCIZ  @AT AN UNEXPECTED @
076153  111  116  103  EMS55:  .ASCIZ  @INCORRECT DATA DURING @
076202  111  116  126  EMS56:  .ASCIZ  @INVALID COMMAND ERROR "IVC" (RMER2, BIT 12) @
076257  104  125  122  EMS57:  .ASCIZ  @DURING DATA TRANSFER @
076305  104  101  124  EMS60:  .ASCIZ  @DATA READ @
076320  104  117  105  EMS61:  .ASCIZ  @DOES NOT COMPARE WITH @
076347  104  101  124  EMS62:  .ASCIZ  @DATA WRITTEN @
076365  042  120  101  EMS63:  .ASCIZ  @"PAR" (RMER1, BIT 3) @
076413  111  123  040  EMS64:  .ASCIZ  @IS INCORRECT @
076431  103  117  115  EMS65:  .ASCIZ  @COMPOSITE ERROR "ERR" (RMDS, BIT 14) @
076477  104  101  124  EMS66:  .ASCIZ  @DATA PARITY ERROR "DPE" (RMER2, BIT 3) @
076547  104  125  122  EMS67:  .ASCIZ  @DURING SEEK COMMAND @
```

```
 58 076574  111  123  040  EMS70:  .ASCIZ  @IS RESET @
 59 076606  105  122  122  EMS71:  .ASCIZ  @ERRONEOUS @
 60 076621  111  123  040  EMS72:  .ASCIZ  @IS SET @
 61 076631  104  111  104  EMS73:  .ASCIZ  @DID NOT SET @
 62 076646  104  111  104  EMS74:  .ASCIZ  @DID NOT RESET @
 63 076665  114  117  123  EMS75:  .ASCIZ  @LOST @
 64 076673  104  125  122  EMS76:  .ASCIZ  @DURING PACK ACK COMMAND @
 65 076724  042  122  115  EMS77:  .ASCIZ  @"RMR" (RMER1, BIT 2) @
 66 076752  042  111  114  EMS100: .ASCIZ  @"ILR" (RMER1, BIT 1) @
 67 077000  042  111  114  EMS101: .ASCIZ  @"ILF" (RMER1, BIT 0) @
 68 077026  104  125  122  EMS102: .ASCIZ  @DURING SEARCH COMMAND @
 69 077055  105  122  122  EMS103: .ASCIZ  @ERROR @
 70 077064  042  114  102  EMS104: .ASCIZ  @"LBC" (RMER2, BIT 10) @
 71 077113  042  114  123  EMS105: .ASCIZ  @"LSC" (RMER2, BIT 11) @
 72 077142  110  105  101  EMS106: .ASCIZ  @HEADER ERRORS @
 73 077161  102  125  123  EMS110: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
 74 077210  101  104  104  EMS111: .ASCIZ  @ADDRESS @
 75 077221  127  110  105  EMS112: .ASCII  @WHEN READING/WRITING RH REGISTERS @
 76 077263  101  124  040          .ASCIZ  @AT THE FOLLOWING @
 77 077305  124  110  105  EMS113: .ASCIZ  @THE SELECTED DEVICE IS @
 78 077335  116  117  116  EMS114: .ASCIZ  @NONEXISTENT DEVICE @
 79 077361  015  012  000  EMS115: .ASCIZ  <CR><LF>
 80 077364  124  110  105  EMS116: .ASCIZ  @THE MASSBUS CONTROLLER @
 81 077414  106  101  111  EMS117: .ASCIZ  @FAILED TO DETECT @
 82 077436  116  117  124  EMS120: .ASCIZ  @NOT AN RM05/3/2 @
 83 077457  103  117  116  EMS121: .ASCIZ  @CONTROL STATUS REGISTER 1, RMCS1, @
 84 077522  102  125  123  EMS122: .ASCIZ  @BUS ADDRESS REGISTER, RMBA, @
 85 077557  103  117  116  EMS123: .ASCIZ  @CONTROL STATUS REGISTER 2, RMCS2, @
 86 077622  105  122  122  EMS124: .ASCIZ  @ERROR REGISTER 1, RMER1, @
 87 077654  101  124  124  EMS125: .ASCIZ  @ATTENTION SUMMARY REGISTER, RMAS, @
 88 077717  115  101  111  EMS126: .ASCIZ  @MAINTENANCE REGISTER #1, RMMR #1, @
 89 077762  105  103  103  EMS127: .ASCIZ  @ECC POSITION REGISTER, RMEC1, @
 90 100021  105  103  103  EMS130: .ASCIZ  @ECC PATTERN REGISTER, RMEC2, @
 91 100057  115  101  111  EMS131: .ASCIZ  @MAINTENANCE REGISTER 2, RMMR2, @
 92 100117  116  117  124  EMS132: .ASCIZ  @NOT INITIALIZED BY @
 93 100143  125  116  111  EMS133: .ASCIZ  @UNIBUS INITIALIZE @
 94 100166  103  117  116  EMS134: .ASCIZ  @CONTROLLER CLEAR, I.E.  BIT 5 OF @
 95 100230  122  110  057  EMS135: .ASCIZ  @RH/RM ERROR CLEAR (RMCS1, BIT 14) @
 96 100273  104  122  111  EMS136: .ASCIZ  @DRIVE CLEAR COMMAND @
 97 100320  104  122  111  EMS137: .ASCIZ  @DRIVE STATUS REGISTER, RMDS @
 98 100355  115  105  104  EMS140: .ASCIZ  @MEDIUM OFF LINE @
 99 100376  042  115  117  EMS141: .ASCIZ  @"MOL" (RMDS, BIT 12) @
100 100424  104  122  111  EMS142: .ASCIZ  @DRIVE FAULT @
101 100441  042  104  126  EMS143: .ASCIZ  @"DVC" (RMER2, BIT 7) @
102 100467  125  116  123  EMS144: .ASCIZ  @UNSAFE @
103 100477  042  125  116  EMS145: .ASCIZ  @"UNS" (RMER1, BIT 14) @
104 100526  123  110  117  EMS146: .ASCIZ  @SHOULD BE SET @
105 100545  117  106  106  EMS147: .ASCIZ  @OFFSET MODE @
106 100562  040  126  117  EMS150: .ASCIZ   @ VOLUME VALID @
107 100601  042  117  115  EMS151: .ASCIZ  @"OM" (RMDS, BIT 0) @
108 100625  042  126  126  EMS152: .ASCIZ  @"VV" (RMDS, BIT 6) @
109 100651  120  101  103  EMS153: .ASCIZ  @PACK ACK COMMAND @
110 100673  116  117  124  EMS154: .ASCIZ  @NOT SET BY @
111 100707  117  106  106  EMS155: .ASCIZ  @OFFSET COMMAND @
112 100727  116  117  124  EMS156: .ASCIZ  @NOT RESET BY @
113 100745  122  124  103  EMS157: .ASCIZ  @RTC COMMAND @
114 100762  122  111  120  EMS160: .ASCIZ  @RIP COMMAND @
```

```
                115 100777      117     106     106  EMS161: .ASCIZ  @OFFSET REGISTER (RMOF) @
                116 101027                           EMS162:         ;<UNUSED>
                117 101027      104     125     122  EMS163: .ASCIZ  @DURING RECALIBRATE @
                118 101053      111     123     040  EMS164: .ASCII  @IS INTERMITTENT OR DRIVE DIDNT DROP ON @
                119 101122      103     131     114          .ASCIZ  @CYLINDER @
                120 101134      125     116     105  EMS165: .ASCIZ  @UNEXPECTED @
                121 101150      122     105     103  EMS166: .ASCIZ  @RECALIBRATE COMMAND @
                122 101175      042     101     124  EMS167: .ASCIZ  @"ATA" (RMDS, BIT15) @
                123 101222      127     110     105  EMS170: .ASCIZ  @WHEN READING REGISTER @
                124 101251      105     122     122  EMS171: .ASCIZ  @ERROR REGISTER #2, RMER2, @
                125 101304      116     117     116  EMS172: .ASCIZ  @NONRECOVERABLE @
                126 101324      122     105     103  EMS173: .ASCIZ  @RECOVERABLE @
                127 101341      104     101     124  EMS174: .ASCIZ  @DATA @
                128 101347      104     125     122  EMS175: .ASCIZ  @DURING WRITE COMMAND @
                129 101375      042     117     120  EMS176: .ASCIZ  @"OPE" (RMER2, BIT 13) @
                130 101424      111     116     040  EMS177: .ASCIZ  @IN WRITE PROTECT @
                131 101446      103     101     116  EMS200: .ASCIZ  @CAN NOT SET @
                132 101463      104     111     101  EMS201: .ASCIZ  @DIAGNOSTIC MODE "DMD" (RMMR1, BIT 0) @
                133 101531      104     125     122  EMS202: .ASCIZ  @DURING DIAGNOSTIC MODE @
                134 101561      111     116     103  EMS203: .ASCIZ  @INCORRECT @
                135 101574      042     127     122  EMS204: .ASCIZ  @"WRL" (RMDS, BIT 11) @
                136 101622      105     130     105  EMS205: .ASCIZ  @EXECUTED @
                137 101634      127     111     124  EMS206: .ASCIZ  @WITH COMP ERROR SET @
                138 101661      042     107     117  EMS207: .ASCIZ  @"GO" (RMCS1, BIT 0) @
                139 101706      127     122     111  EMS210: .ASCIZ  @WRITING @
                140 101717      127     101     123  EMS211: .ASCIZ  @WAS RESET BY @
                141 101735      120     122     117  EMS212: .ASCIZ  @PROGRAM INTERRUPT @
                142 101760      127     101     123  EMS213: .ASCIZ  @WAS NOT GENERATED @
                143 102003      123     105     105  EMS214: .ASCIZ  @SEEK COMMAND @
                144 102021      120     122     117  EMS215: .ASCIZ  @PROGRAM TIMEOUT @
                145 102042      104     125     122  EMS216: .ASCIZ  @DURING LOOK AHEAD TEST @
                146 102072      114     117     117  EMS217: .ASCIZ  @LOOK AHEAD REGISTER,RMLA, @
                147 102125      123     105     101  EMS220: .ASCIZ  @SEARCH COMMAND @
                148 102145      102     101     104  EMS221: .ASCIZ  @BAD SECTOR ERROR "BSE" (RMER2, BIT 15) @
                149 102215      101     040     104  EMS222: .ASCIZ  @A DATA TRANSFER COMMAND @
                150 102246      110     105     101  EMS223: .ASCIZ  @HEADER COMPARE INHIBIT "HCI" (RMOF, BIT 10) @
                151 102323      116     117     116  EMS224: .ASCIZ  @NONEXISTENT MEMORY "NEM" (RMCS2, BIT 11) @
                152
```

```
 1 102375  105  130  120  EH1:    .ASCIZ  @EXPCTD  RECEVD@
 2 102414  102  125  123  EH110:  .ASCIZ  @BUSADR@
 3 102423  040  122  115  EH111:  .ASCIZ  @ RMCS2   RMCS1@
 4
 5 102442  122  105  103  EH114:  .ASCIZ  @RECEVD  SNGPRT  DULPRT@
 6 102471  105  130  120  EH223:  .ASCIZ  @EXPCTD  RECEVD   DATA@
 7 102517  105  130  120  EH256:  .ASCII  @EXPCTD  RECEVD  RGSTR@<CRLF>
 8 102545  123  124  101          .ASCIZ  @STATUS  STATUS  INDEX@
 9
10 102573  107  104  101  EH336:  .ASCIZ  @GDADRS  GDDATA  BDADRS  BDDATA@
11 102632  122  115  103  EH337:  .ASCII  @RMCS2   STATUS  FAILING   DATA@<CRLF>
12 102671  137  137  137          .ASCII  @                          @<CRLF>
13 102730  105  130  120          .ASCIZ  @EXPCTD--RECEVD  --BIT---ADRESS@
14
15 102767  122  115  105  EH344:  .ASCII  @RMER1   STATUS  HEADER  FAILING@<CRLF>
16 103027  137  137  137          .ASCII  @                 WORD     BIT@<CRLF>
17 103065  105  130  120          .ASCIZ  @EXPCTD--RECEVD  NUMBER  POSITON@
18 103125  105  130  120  EH353:  .ASCII  @EXPCTD  RECEVD@<CRLF>
19 103144  040  122  115          .ASCIZ  @ RMLA    RMLA    RMOF @
20
21 103173  040  122  115  STSH1:  .ASCII  @ RMCS1  RMCS2   RMDS    RMER1   RMER2@
22 103240  040  040  040          .ASCIZ  @   RMAS@
23 103250  040  122  115  STSH2:  .ASCII  @ RMWC    RMBA    RMDA    RMOF    RMDC@
24 103315  040  040  040          .ASCIZ  @    RMEC1   RMEC2@
25 103337  040  122  115  STSH3:  .ASCIZ  @ RMDA    RMDC    RMOF    RMLA@
26 103375  040  122  115  STSH4:  .ASCIZ  @ RMMR1   RMMR2   RMDT    RMSN@
27                        .EVEN
28
```

```
 1 103434  001140  001142  000000  ED1:    .WORD   $GDDAT,$BDDAT,0
 2 103442  001276  000000          ED110:  .WORD   $BASE,0
 3 103446  001174  001176  000000  ED111:  .WORD   $TMP0,$TMP1,0
 4
 5 103454  001362  001176  001200  ED114:  .WORD   RMDII,$TMP1,$TMP2,0
 6 103464  001140  001142  001174  ED223:  .WORD   $GDDAT,$BDDAT,$TMP0,0
 7
 8 103474  001134  001140  001136  ED336:  .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
 9
10 103506  001140  001142  001174  ED337:  .WORD   $GDDAT,$BDDAT,$TMP0,$TMP1,0
11 103520  001140  001142  001442  ED353:  .WORD   $GDDAT,$BDDAT,RMOFO,0
12
13 103530  001334  001344  001346  STSD1:  .WORD   RMCS1I,RMCS2I,RMDSI,RMER1I,RMER2I,RMASI,0
14 103546  001336  001340  001342  STSD2:  .WORD   RMWCI,RMBAI,RMDAI,RMOFI,RMDCI,RMEC1I
15 103562  001402  000000          .WORD   RMEC2I,0
16 103566  001342  001370  001366  STSD3:  .WORD   RMDAI,RMDCI,RMOFI,RMLAI,0
17 103600  001360  001374  001362  STSD4:  .WORD   RMMR1I,RMMR2I,RMDTI,RMSNI,0
18
```

ERROR MESSAGE STRINGS

SEQ 0277

```
        1 103612      000                     EF110:  .BYTE   0
        2 103613      000       000           EF111:  .BYTE   0,0
        3 103615      000       000       000 EF114:  .BYTE   0,0,0
        4 103620      000       000       000 EF336:  .BYTE   0,0,0,0
        5 103624      000       000       000 EF337:  .BYTE   0,0,0,0,0
        6
        7 103631      000       000       000 STSF:   .BYTE   0,0,0,0,0,0,0
        8
```

```
                                BUFFER:
                                BUFONE: .BLKW   258.
                                BUFTWO: .BLKW   258.

                103640                  .=BUFFER

                                HELP:
                                .ASCII  <CRLF>
                                .ASCII  <CRLF>
                                .ASCII  @LIST OF TESTS@<CRLF>
                                .ASCII  @------------@<CRLF>
                                .ASCII  @T1     CONTROLLER ACCESS TEST@<CRLF>
                                .ASCII  @T2     DEVICE AVAILABLE TEST@<CRLF>
                                .ASCII  @T3     DRIVE TYPE TEST@<CRLF>
                                .ASCII  @T4     UNIBUS INITIALIZE TEST@<CRLF>
                                .ASCII  @T5     CONTROLLER CLEAR TEST@<CRLF>
                                .ASCII  @T6     ERROR CLEAR TEST@<CRLF>
                                .ASCII  @T7     DRIVE STATUS TEST@<CRLF>
                                .ASCII  @T10    PRIMARY/SECONDARY ERROR TEST@<CRLF>
                                .ASCII  @T11    DIAGNOSTIC MODE TEST@<CRLF>
                                .ASCII  @T12    PACK ACKNOWLEDGE TEST@<CRLF>
                                .ASCII  @T13    RECALIBRATE TEST@<CRLF>
                                .ASCII  @T14    ABORT RECALIBRATE TEST@<CRLF>
                                .ASCII  @T15    IVC RECALIBRATE TEST@<CRLF>
                                .ASCII  @T16    IAE RECALIBRATE TEST@<CRLF>
                                .ASCII  @T17    RECALIBRATE AT OFFSET@<CRLF>
                                .ASCII  @T20    DRIVE CLEAR TEST@<CRLF>
                                .ASCII  @T21    NOP TEST@<CRLF>
                                .ASCII  @T22    OFFSET TEST@<CRLF>
                                .ASCII  @T23    GO/ATA TEST@<CRLF>
                                .ASCII  @T24    WRITE ATA TEST@<CRLF>
                                .ASCII  @T25    ERROR/ATA TEST@<CRLF>
                                .ASCII  @T26    PROGRAM INTERRUPT TEST@<CRLF>
                                .ASCII  @T27    INHIBIT INTERRUPT TEST@<CRLF>
                                .ASCII  @T30    RETURN TO CENTERLINE TEST@<CRLF>
                                .ASCII  @T31    READ IN PRESET TEST@<CRLF>
                                .ASCII  @T32    RMDC CLEAR OFFSET TEST@<CRLF>
                                .ASCII  @T33    ILLEGAL FUNCTION TEST@<CRLF>
                                .ASCII  @T34    INVALID COMMAND TEST@<CRLF>
                                .ASCII  @T35    INVALID ADDRESS ERROR TEST@<CRLF>
                                .ASCII  @T36    WRITE LOCK ERROR TEST@<CRLF>
                                .ASCII  @T37    ERROR ABORT TESTS@<CRLF>
                                .ASCII  @T40    RMR TEST@<CRLF>
                                .ASCII  @T41    PARITY ERROR TEST@<CRLF>
                                .ASCII  @T42    ILLEGAL REGISTER TEST@<CRLF>
                                .ASCII  @T43    SEEK LAST CYLINDER@<CRLF>
                                .ASCII  @T44    SEEK FIRST CYLINDER@<CRLF>
                                .ASCII  @T45    SEEK PRIME CYLINDERS@<CRLF>
                                .ASCII  @T46    SEEK ZERO DIFFERENCE@<CRLF>
                                .ASCII  @T47    SEEK MAXIMUM DIFFERENCE FORWARD@<CRLF>
                                .ASCII  @T50    SEEK ADJACENT FORWARD@<CRLF>
                                .ASCII  @T51    SEEK ADJACENT REVERSE@<CRLF>
                                .ASCII  @T52    SEEK INVALID SECTOR@<CRLF>
                                .ASCII  @T53    SEEK INVALID TRACK@<CRLF>
                                .ASCII  @T54    SEEK INVALID CYLINDER@<CRLF>
                                .ASCII  @T55    IVC SEEK TEST@<CRLF>
                                .ASCII  @T56    ABORT SEEK TEST@<CRLF>
```

```
     58 106006      124      065      067  .ASCII  @T57    SEEK AT OFFSET@<CRLF>
     59 106031      124      066      060  .ASCII  @T60    LOOK AHEAD TEST@<CRLF>
     60 106055      124      066      061  .ASCII  @T61    SEARCH ON CYLINDER@<CRLF>
     61 106104      124      066      062  .ASCII  @T62    SEARCH OFF CYLINDER@<CRLF>
     62 106134      124      066      063  .ASCII  @T63    SEARCH INVALID SECTOR@<CRLF>
     63 106166      124      066      064  .ASCII  @T64    SEARCH INVALID TRACK@<CRLF>
     64 106217      124      066      065  .ASCII  @T65    SEARCH INVALID CYLINDER@<CRLF>
     65 106253      124      066      066  .ASCII  @T66    IVC SEARCH TEST@<CRLF>
     66 106277      124      066      067  .ASCII  @T67    ABORT SEARCH TEST@<CRLF>
     67 106325      124      067      060  .ASCII  @T70    SEARCH AT OFFSET@<CRLF>
     68 106352      124      067      061  .ASCII  @T71    HEAD ALIGNMENT SEEK@<CRLF>
     69 106402      200               .ASCII  <CRLF>
     70 106403      117      120      105  .ASCII  @OPERATIONAL SWITCH SETTINGS@<CRLF>
     71 106437      055      055      055  .ASCII  @-------------------------@<CRLF>
     72 106473      123      127      111  .ASCII  @SWITCH                  USE@<CRLF>
     73 106510      055      055      055  .ASCII  @------          --------------------@<CRLF>
     74 106545      040      040      061  .ASCII  @  15            HALT ON ERROR@<CRLF>
     75 106571      040      040      061  .ASCII  @  14            LOOP ON TEST@<CRLF>
     76 106614      040      040      061  .ASCII  @  13            INHIBIT ERROR TYPEOUTS@<CRLF>
     77 106651      040      040      061  .ASCII  @  12            @<CRLF>
     78 106660      040      040      061  .ASCII  @  11            INHIBIT ITERATIONS@<CRLF>
     79 106711      040      040      061  .ASCII  @  10            BELL ON ERROR@<CRLF>
     80 106735      040      040      040  .ASCII  @   9            LOOP ON ERROR@<CRLF>
     81 106761      040      040      040  .ASCII  @   8            LOOP ON TEST IN SWR<7:0>@<CRLF>
     82 107020      040      040      040  .ASCII  @   7            TN128@<CRLF>
     83 107034      040      040      040  .ASCII  @   6            TN64@<CRLF>
     84 107047      040      040      040  .ASCII  @   5            TN32@<CRLF>
     85 107062      040      040      040  .ASCII  @   4            TN16@<CRLF>
     86 107075      040      040      040  .ASCII  @   3            TN8@<CRLF>
     87 107107      040      040      040  .ASCII  @   2            TN4@<CRLF>
     88 107121      040      040      040  .ASCII  @   1            TN2@<CRLF>
     89 107133      040      040      040  .ASCIZ  @   0            TN1@<CRLF>
     90
     91          000200                   .END    200
```

SEQ 0280

```
ABASE = 176700          AUNIT = 000000          CLKVCT  001512          EDT223  074046          EMS105  077113
ACDW1 = 000000          AUSWR = 000000          CLR    = 000040          EDT336  074056          EMS106  077142
ACDW2 = 000000          AVECT1= 120254          CLRSTS  053646          EDT337  074066          EMS11   074576
ACKSTS  054526          AVECT2= 000000          CMNSTA  007604          EDT344  074076          EMS110  077161
ACPUOP= 000000          A16    = 000400         CMPERR  051644          EDT353  074110          EMS111  077210
ADDW0 = 000000          A17    = 001000         CNSL00  066304          ED1     103434          EMS112  077221
ADDW1 = 000000          BACK   = 000001         CNSL01  066337          ED110   103442          EMS113  077305
ADDW10= 000000          BADTMO  005332          CNSL02  066355          ED111   103446          EMS114  077335
ADDW11= 000000          BAI    = 000010         CNSL03  066417          ED114   103454          EMS115  077361
ADDW12= 000000          BB00   = 000001         CNSL04  066437          ED223   103464          EMS116  077364
ADDW13= 000000          BB01   = 000002         CNSL05  066473          ED336   103474          EMS117  077414
ADDW14= 000000          BB02   = 000004         CNSL06  066505          ED337   103506          EMS12   074607
ADDW15= 000000          BB03   = 000010         CNSL07  066536          ED353   103520          EMS120  077436
ADDW2 = 000000          BB04   = 000020         CNSL08  066701          EECC   = 000020          EMS121  077457
ADDW3 = 000000          BB05   = 000040         CNSL09  066702          EFT1    074112          EMS122  077522
ADDW4 = 000000          BB06   = 000100         CNTCLR  053530          EFT110  074130          EMS123  077557
ADDW5 = 000000          BB07   = 000200         COMMA   066215          EFT111  074132          EMS124  077622
ADDW6 = 000000          BB08   = 000400         CONT   = 000100         EFT114  074134          EMS125  077654
ADDW7 = 000000          BB09   = 001000         CR     = 000015         EFT2    074122          EMS126  077717
ADDW8 = 000000          BIT0   = 000001         CRLF   = 000200         EFT223  074136          EMS127  077762
ADDW9 = 000000          BIT00  = 000001         CTLFG   001326          EFT336  074146          EMS13   074650
ADEVCT= 000000          BIT01  = 000002         CYLMSK= 001777          EFT337  074156          EMS130  100021
ADEVM = 000000          BIT02  = 000004         DBCK   = 100000         EFT344  074166          EMS131  100057
ADR    = 000001         BIT03  = 000010         DBEN   = 040000         EFT353  074176          EMS132  100117
AENV   = 000000         BIT04  = 000020         DBL    = 002000         EF110   103612          EMS133  100143
AENVM = 000000          BIT05  = 000040         DCK    = 100000         EF111   103613          EMS134  100166
AFATAL= 000000          BIT06  = 000100         DDISP  = 177570         EF114   103615          EMS135  100230
ALL     066204          BIT07  = 000200         DEBL   = 020000         EF336   103620          EMS136  100273
AMADR1= 000000          BIT08  = 000400         DEVSEL  052056          EF337   103624          EMS137  100320
AMADR2= 000000          BIT09  = 001000         DISPLA  001156          EHT1    073676          EMS14   074661
AMADR3= 000000          BIT1   = 000002         DISPRE  000174          EHT110  073720          EMS140  100355
AMADR4= 000000          BIT10  = 002000         DLT    = 100000         EHT111  073724          EMS141  100376
AMAMS1= 000000          BIT11  = 004000         DMD    = 000001         EHT114  073730          EMS142  100424
AMAMS2= 000000          BIT12  = 010000         DPE    = 000010         EHT2    073710          EMS143  100441
AMAMS3= 000000          BIT13  = 020000         DPEHI  = 040000         EHT223  073734          EMS144  100467
AMAMS4= 000000          BIT14  = 040000         DPELO  = 020000         EHT256  073746          EMS145  100477
AMSGAD= 000000          BIT15  = 100000         DPR    = 000400         EHT336  073760          EMS146  100526
AMSGLG= 000000          BIT2   = 000004         DRQ    = 004000         EHT337  073772          EMS147  100545
AMSGTY= 000000          BIT3   = 000010         DRVCLR= 000010          EHT344  074004          EMS15   074672
AMTYP1= 000000          BIT4   = 000020         DRVSTS  057064          EHT353  074016          EMS150  100562
AMTYP2= 000000          BIT5   = 000040         DRY    = 000200         EH1     102375          EMS151  100601
AMTYP3= 000000          BIT6   = 000100         DSWR   = 177570         EH110   102414          EMS152  100625
AMTYP4= 000000          BIT7   = 000200         DTE    = 010000         EH111   102423          EMS153  100651
AOE    = 001000         BIT8   = 000400         DTO    = 010000         EH114   102442          EMS154  100673
APASS  = 000000         BIT9   = 001000         DULPRT= 024024          EH223   102471          EMS155  100707
APE    = 100000         BOTADR  043106          DVA    = 004000         EH256   102517          EMS156  100727
APRIOR= 000000          BOTFLG  043110          DVC    = 000200         EH336   102573          EMS157  100745
APTCSU= 000040          BPTVEC= 000014          EARLY   067472          EH337   102632          EMS16   074714
APTENV= 000001          BSE    = 100000         EBL    = 020000         EH344   102767          EMS160  100762
APTSIZ= 000200          BUFFER  103640          ECH    = 000100         EH353   103125          EMS161  100777
APTSPO= 000100          BUFONE  103640          ECI    = 004000         EMS1    074200          EMS162  101027
ARGS   = 000002         BUFTWO  104644          ECRC   = 001000         EMS10   074534          EMS163  101027
ASWREG= 000000          CC     = 004000         EDT1    074022          EMS100  076752          EMS164  101053
ATA    = 100000         CH     = 002000         EDT110  074040          EMS101  077000          EMS165  101134
ATESTN= 000000          CHRCNT  043111          EDT111  074042          EMS102  077026          EMS166  101150
ATNMSK= 000377          CKSWR  = 104410         EDT114  074044          EMS103  077055          EMS167  101175
ATNTBL  067146          CLKADR  001510          EDT2    074032          EMS104  077064          EMS17   074744
```

SYMBOL TABLE

EMS170	101222	EMS50	076032	EMT131	071224	EMT213	072130	EMT276	072730
EMS171	101251	EMS51	076061	EMT132	071236	EMT214	072140	EMT277	072744
EMS172	101304	EMS52	076107	EMT133	071250	EMT215	072160	EMT3	067616
EMS173	101324	EMS53	076117	EMT134	071262	EMT216	072172	EMT30	070040
EMS174	101341	EMS54	076131	EMT135	071274	EMT217	072202	EMT300	072762
EMS175	101347	EMS55	076153	EMT136	071306	EMT22	067760	EMT301	073002
EMS176	101375	EMS56	076202	EMT137	071320	EMT220	072212	EMT302	073024
EMS177	101424	EMS57	076257	EMT14	067704	EMT221	072222	EMT303	073036
EMS2	074247	EMS6	074422	EMT140	071330	EMT222	072232	EMT304	073050
EMS20	075004	EMS60	076305	EMT141	071340	EMT223	072242	EMT305	073062
EMS200	101446	EMS61	076320	EMT142	071350	EMT224	072252	EMT306	073074
EMS201	101463	EMS62	076347	EMT143	071360	EMT225	072254	EMT307	073104
EMS202	101531	EMS63	076365	EMT144	071370	EMT226	072256	EMT31	070050
EMS203	101561	EMS64	076413	EMT145	071400	EMT227	072260	EMT310	073124
EMS204	101574	EMS65	076431	EMT146	071410	EMT23	067770	EMT311	073136
EMS205	101622	EMS66	076477	EMT147	071420	EMT230	072262	EMT312	073150
EMS206	101634	EMS67	076547	EMT15	067712	EMT231	072264	EMT313	073162
EMS207	101661	EMS7	074467	EMT150	071430	EMT232	072266	EMT314	073202
EMS21	075027	EMS70	076574	EMT151	071440	EMT233	072270	EMT315	073214
EMS210	101706	EMS71	076606	EMT152	071452	EMT234	072272	EMT316	073226
EMS211	101717	EMS72	076621	EMT153	071464	EMT235	072274	EMT317	073240
EMS212	101735	EMS73	076631	EMT154	071502	EMT236	072276	EMT32	070060
EMS213	101760	EMS74	076646	EMT155	071520	EMT237	072300	EMT320	073250
EMS214	102003	EMS75	076665	EMT156	071532	EMT24	070000	EMT321	073260
EMS215	102021	EMS76	076673	EMT157	071544	EMT240	072302	EMT322	073270
EMS216	102042	EMS77	076724	EMT16	067720	EMT241	072304	EMT323	073276
EMS217	102072	EMTVEC=	000030	EMT160	071556	EMT242	072306	EMT324	073306
EMS22	075052	EMT1	067604	EMT161	071566	EMT243	072310	EMT325	073320
EMS220	102125	EMT10	067654	EMT162	071600	EMT244	072312	EMT326	073332
EMS221	102145	EMT100	070652	EMT163	071612	EMT245	072314	EMT327	073350
EMS222	102215	EMT101	070670	EMT164	071622	EMT246	072316	EMT33	070070
EMS223	102246	EMT102	070706	EMT165	071630	EMT247	072326	EMT330	073362
EMS224	102323	EMT103	070716	EMT166	071636	EMT25	070010	EMT331	073372
EMS23	075066	EMT104	070730	EMT167	071644	EMT250	072340	EMT332	073404
EMS24	075114	EMT105	070746	EMT17	067730	EMT251	072354	EMT333	073416
EMS25	075143	EMT106	070756	EMT170	071652	EMT252	072362	EMT334	073434
EMS26	075160	EMT107	070766	EMT171	071662	EMT253	072370	EMT335	073452
EMS27	075201	EMT11	067660	EMT172	071672	EMT254	072406	EMT336	073472
EMS3	074264	EMT110	071006	EMT173	071702	EMT255	072416	EMT337	073502
EMS30	075212	EMT111	071020	EMT174	071712	EMT256	072426	EMT34	070100
EMS31	075222	EMT112	071026	EMT175	071724	EMT257	072436	EMT340	073512
EMS32	075251	EMT113	071034	EMT176	071732	EMT26	070020	EMT341	073524
EMS33	075300	EMT114	071050	EMT177	071740	EMT260	072450	EMT342	073532
EMS34	075326	EMT115	071056	EMT2	067610	EMT261	072462	EMT343	073544
EMS35	075377	EMT116	071066	EMT20	067740	EMT262	072502	EMT344	073556
EMS36	075426	EMT117	071076	EMT200	071752	EMT263	072512	EMT345	073570
EMS37	075455	EMT12	067664	EMT201	071764	EMT264	072522	EMT346	073600
EMS4	074327	EMT120	071106	EMT202	071776	EMT265	072542	EMT347	073614
EMS40	075503	EMT121	071116	EMT203	072010	EMT266	072562	EMT35	070110
EMS41	075532	EMT122	071126	EMT204	072022	EMT267	072574	EMT350	073626
EMS42	075560	EMT123	071136	EMT205	072040	EMT27	070030	EMT351	073644
EMS43	075607	EMT124	071146	EMT206	072050	EMT270	072612	EMT352	073654
EMS44	075636	EMT125	071156	EMT207	072060	EMT271	072626	EMT353	073660
EMS45	075711	EMT126	071166	EMT21	067750	EMT272	072644	EMT354	073670
EMS46	075754	EMT127	071200	EMT210	072072	EMT273	072662	EMT36	070120
EMS47	076003	EMT13	067672	EMT211	072100	EMT274	072700	EMT37	070130
EMS5	074372	EMT130	071212	EMT212	072114	EMT275	072716	EMT4	067624

```
EMT40   070140      F4     = 000040      LS     = 000004      PR2    = 000100      RMDS   = 000012
EMT41   070146      GET      044154      LSC    = 004000      PR3    = 000140      RMDSI    001346
EMT42   070156      GETBUF   001334      LST    = 000002      PR4    = 000200      RMDSO    001422
EMT43   070170      GETINX   001514      LSTRK    001332      PR5    = 000240      RMDT   = 000026
EMT44   070200      GETSTS   044070      MCLK   = 004000      PR6    = 000300      RMDTI    001362
EMT45   070210      GO     = 000001      MCPE   = 020000      PR7    = 000340      RMDTO    001436
EMT46   070220      GTSWR  = 104407      MDF    = 000100      PS     = 177776      RMEC1  = 000044
EMT47   070230      HCE    = 000200      MDPE   = 000400      PSEL   = 002000      RMEC1I   001400
EMT5    067632      HCI    = 002000      MI     = 000004      PSW    = 177776      RMEC1O   001454
EMT50   070236      HCRC   = 000400      MIXED    067156      PUT      044424      RMEC2  = 000046
EMT51   070246      HELP     103640      MOC    = 000400      PUTBUF   001410      RMEC2I   001402
EMT52   070260      HT     = 000011      MOH    = 020000      PUTINX   001543      RMEC2O   001456
EMT53   070276      IAE    = 002000      MOL    = 010000      PWRVEC=  000024      RMER1  = 000014
EMT54   070314      IDXMSK = 000077      MRD    = 002000      QUES     066211      RMER1I   001350
EMT55   070324      IE     = 000100      MS     = 000040      RCLSTS   055322      RMER1O   001424
EMT56   070336      ILF    = 000001      MSC    = 000002      RD     = 000070      RMER2  = 000042
EMT57   070354      ILF02  = 000002      MSDRVS   066723      RDCHR  = 104411      RMER2I   001376
EMT6    067640      ILF24  = 000024      MSE    = 100000      RDLIN  = 104412      RMER2O   001452
EMT60   070364      ILF26  = 000026      MSER   = 000200      RDOCT  = 104413      RMHR   = 000036
EMT61   070402      ILF30  = 000030      MSGDRV   066737      RDY    = 000200      RMHR1    001372
EMT62   070422      ILF32  = 000032      MSHELP   066220      READY    007756      RMHRO    001446
EMT63   070436      ILF34  = 000034      MUR    = 001000      RECAL  = 000006      RMLA   = 000020
EMT64   070440      ILF36  = 000036      MWD    = 000010      RESREG=  104415      RMLAI    001354
EMT65   070460      ILF40  = 000040      MWP    = 000010      RESVEC=  000010      RMLAO    001430
EMT66   070500      ILF42  = 000042      MXF    = 001000      REX    = 010000      RMMR1  = 000024
EMT67   070512      ILF44  = 000044      NDTMSK = 115760      RG     = 040000      RMMR1I   001360
EMT7    067646      ILF46  = 000046      NED    = 010000      RGDTPT   067156      RMMR1O   001434
EMT70   070524      ILF54  = 000054      NEM    = 004000      RH     = 000072      RMMR2  = 000040
EMT71   070534      ILF56  = 000056      NOP    = 000000      RIP    = 000020      RMMR2I   001374
EMT72   070544      ILF64  = 000064      NOTAVL   067002      RLEASE=  000012      RMMR2O   001450
EMT73   070562      ILF66  = 000066      NOTPRS   066765      RMAS   = 000016      RMOF   = 000032
EMT74   070600      ILF74  = 000074      NSA    = 100000      RMASI    001352      RMOFI    001366
EMT75   070610      ILF76  = 000076      OCC    = 100000      RMASO    001426      RMOFO    001442
EMT76   070630      ILR    = 000002      OFD    = 000200      RMBA   = 000004      RMR    = 000004
EMT77   070640      ILRG50 = 000050      OFFLIN   067021      RMBAE  = 000050      RMSN   = 000030
ENRGDT  067604      ILRG52 = 000052      OFFSET=  000014      RMBAEI   001404      RMSNI    001364
EQUALS  066202      ILRG54 = 000054      OM     = 000001      RMBAEO   001460      RMSNO    001440
ERR    = 040000     ILRG56 = 000056      ONES     067216      RMBAI    001340      RMWC   = 000002
ERRNMB  043104      ILRG60 = 000060      ONLINE   067033      RMBAO    001414      RMWCI    001336
ERROR  = 104000     ILRG62 = 000062      OPE    = 020000      RMCS1  = 000000      RMWCO    001412
ERRTYP  042346      ILRG64 = 000064      OPI    = 020000      RMCS1I   001334      RQA    = 100000
ERRVEC= 000004      ILRG66 = 000066      OR     = 000200      RMCS1O   001410      RQB    = 040000
ERTY00  043112      ILRG70 = 000070      PACACK = 000022      RMCS2  = 000010      RTC    = 000016
ERTY01  043120      ILRG72 = 000072      PAKACK = 000022      RMCS2I   001344      R6     =%000006
ERTY02  043130      ILRG74 = 000074      PAR    = 000010      RMCS2O   001420      R7     =%000007
ERTY03  043137      ILRG76 = 000076      PAT    = 000020      RMCS3  = 000052      SADMSK=  000377
ERTY04  043145      IOTVEC = 000020      PDA    = 000400      RMCS3I   001406      SAVREG=  104414
ESRC   = 004000     IPCK0  = 000001      PGE    = 002000      RMCS3O   001462      SA1    = 000001
FER    = 000020     IPCK1  = 000002      PGM    = 001000      RMDA   = 000006      SA16   = 000020
FIND   = 000001     IPCK2  = 000004      PHA    = 000200      RMDAI    001342      SA2    = 000002
FMT16  = 010000     IPCK3  = 000010      PIP    = 020000      RMDAO    001416      SA4    = 000004
FNCDTB  067046      IR     = 000100      PIRQ   = 177772      RMDB   = 000022      SA8    = 000010
FNCMSK= 000077      IVC    = 010000      PIRQVE=  000240      RMDBI    001356      SC     = 100000
F0     = 000002     LBC    = 002000      PLFS   = 002000      RMDBO    001432      SCHSTS   057666
F1     = 000004     LBT    = 002000      PRIERR   045152      RMDC   = 000034      SCOPE  = 000004
F2     = 000010     LF     = 000012      PR0    = 000000      RMDCI    001370      SCTMSK=  003700
F3     = 000020     LODEV    066745      PR1    = 000040      RMDCO    001444      SCO    - 000100
```

```
SC1    = 000200        TAGADR= 001114        TST52   032172        $BIN    062200        $ITEMB  001130
SC2    = 000400        TAP    = 040000        TST53   032610        $CDW1   001302        $LF     001220
SC3    = 001000        TA1    = 000400        TST54   033236        $CDW2   001304        $LFLG   066177
SC4    = 002000        TA16   = 010000        TST55   033654        $CHARC  063204        $LPADR  001122
SEARCH= 000030         TA2    = 001000        TST56   034214        $CKSWR  064406        $LPERR  001124
SECERR  046004         TA4    = 002000        TST57   034556        $CMTAG  001114        $MADR1  001254
SEEK   = 000004        TA8    = 004000        TST6    012076        $CM3   = 000000        $MADR2  001260
SEKSTS  052270         TBITVE= 000014         TST60   035110        $CM4   = 000005        $MADR3  001264
SHUT    061766         TIMOUT  044766         TST61   035536        $CNTLC  065304        $MADR4  001270
SIZCLK  044644         TKVEC  = 000060        TST62   036214        $CNTLG  065316        $MAIL   001222
SKI    = 040000        TPVEC  = 000064        TST63   036720        $CNTLU  065311        $MAMS1  001252
SNGPRT= 020024         TRAPVE= 000034         TST64   037320        $CPUOP  001250        $MAMS2  001256
STACK  = 001100        TRE    = 040000        TST65   037730        $CRLF   001217        $MAMS3  001262
STANDA  006700         TRTVEC= 000014         TST66   040330        $DBLK   062416        $MAMS4  001266
START   005412         TST    = 010000        TST67   040710        $DDW0   001306        $MBADR  001102
STCDRV  061232         TSTNMB  043102         TST7    012302        $DDW1   001310        $MFLG   066176
STKLMT= 177774         TSTPRP  043150         TST70   041254        $DDW2   001312        $MNEW   065334
STOP    061736         TSTQUE  001464         TST71   041612        $DDW3   001314        $MSGAD  001236
STSD1   103530         TST1    010116         TYPBN  = 104406        $DDW4   001316        $MSGLG  001240
STSD2   103546         TST10   012616         TYPDS  = 104405        $DDW5   001320        $MSGTY  001222
STSD3   103566         TST11   012776         TYPE   = 104401        $DDW6   001322        $MSWR   065323
STSD4   103600         TST12   014050         TYPOC  = 104402        $DDW7   001324        $MTYP1  001253
STSF    103631         TST13   014350         TYPON  = 104404        $DEVCT  001232        $MTYP2  001257
STSH1   103173         TST14   014566         TYPOS  = 104403        $DEVM   001300        $MTYP3  001263
STSH2   103250         TST15   015074         UBUSQS  066251        $DOAGN  042336        $MTYP4  001267
STSH3   103337         TST16   015404         UNS    = 040000        $DTBL   062406        $MXCNT  063524
STSH4   103375         TST17   015646         UNTMSK= 000007        $ENDAD  042326        $NULL   001170
SWR     001154         TST2    010302         UPE    = 020000        $ENDCT  042172        $NWTST= 000001
SWREG   000176         TST20   016124         USE    = 040000        $ENULL  042342        $OCNT   062650
SW0    = 000001        TST21   016374         U0     = 000001        $ENV    001242        $OMODE  062652
SW00   = 000001        TST22   016624         U1     = 000002        $ENVM   001243        $OVER   063510
SW01   = 000002        TST23   017150         U2     = 000004        $EOP    042136        $PASS   001230
SW02   = 000004        TST24   017570         VV     = 000100        $EOPCT  042164        $PASTM  001106
SW03   = 000010        TST25   020134         WC     = 000040        $EOSP   042100        $POWER  065724
SW04   = 000020        TST26   020526         WCD    = 000050        $ERFLG  001117        $PWRDN  065556
SW05   = 000040        TST27   021216         WCE    = 040000        $ERMAX  001131        $PWRMG  065712
SW06   = 000100        TST3    010466         WCEHI  = 010000        $ERROR  063710        $PWRUP  065630
SW07   = 000200        TST30   021564         WCELO  = 004000        $ERRPC  001132        $QUES   001216
SW08   = 000400        TST31   022106         WCF    = 000040        $ERRTB  001572        $RDCHR  064750
SW09   = 001000        TST32   022470         WCH    = 000052        $ERTTL  001126        $RDLIN  065040
SW1    = 000002        TST33   023006         WD     = 000060        $ESCAP  001210        $RDOCT  065346
SW10   = 002000        TST34   023572         WH     = 000062        $ETABL  001242        $RDSZ  = 000010
SW11   = 004000        TST35   024316         WLE    = 004000        $ETEND  001326        $RESRE  062070
SW12   = 010000        TST36   025106         WRL    = 004000        $FATAL  001224        $RTNAD  042340
SW13   = 020000        TST37   025672         XSIZ    006454        $FFLG   066200        $SAVRE  062032
SW14   = 040000        TST4    010670         XXDP    001330        $FILLC  001172        $SAVR6  065722
SW15   = 100000        TST40   026224         ZEROS   067260        $FILLS  001171        $SCOPE  063210
SW2    = 000004        TST41   026664         $APTHD  001100        $GDADR  001134        $SETUP= 000137
SW3    = 000010        TST42   027104         $ATYC   065760        $GDDAT  001140        $STUP  = 177777
SW4    = 000020        TST43   027442         $ATY1   065734        $GET42  042316        $SVLAD  063454
SW5    = 000040        TST44   027730         $ATY3   065742        $GTSWR  064476        $SVPC  = 000204
SW6    = 000100        TST45   030216         $ATY4   065752        $HD    = 000000        $SWR   = 167400
SW7    = 000200        TST46   030530         $AUTOB  001150        $HIBTS  001100        $SWREG  001244
SW8    = 000400        TST47   031036         $BASE   001276        $HIOCT  065446        $SWRMK= 000000
SW9    = 001000        TST5    011540         $BDADR  001136        $ICNT   001120        $SW08T  063526
TADMSK= 177400         TST50   031350         $BDDAT  001142        $ILLUP  065716        $TESTN  001226
TAG    = 020000        TST51   031660         $BELL   001212        $INTAG  001151        $TIMES  001206
```

```
$TKB    001162          $TMP0   001174          $TRAP   065450          $TYPE   062654          $VECT1  001272
$TKCNT  064110          $TMP1   001176          $TRAP2  065510          $TYPEC  063066          $VECT2  001274
$TKINT  064120          $TMP2   001200          $TRP    = 000016        $TYPEX  063206          $XOFF   = 000023
$TKQEN= 064117          $TMP3   001202          $TRPAD  065522          $TYPOC  062452          $XON    = 000021
$TKQIN  064112          $TMP4   001204          $TSTM   001104          $TYPON  062466          $XTSTR  063226
$TKQOU  064114          $TN     = 000072        $TSTNM  001116          $TYPOS  062426          $$GET4= 000000
$TKQSR  064116          $TPB    001166          $TTYIN  065274          $UNIT   001234          $$SW08= 000072
$TKS    001160          $TPFLG  001173          $TYPBN  062126          $UNITM  001110          $OFILL  062651
$TKSRV  064170          $TPS    001164          $TYPDS  062202          $USWR   001246          .$X     = 001100

. ABS.  107146      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  55208 WORDS  ( 216 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  69 PAGES
CZRMMA.BIC,CZRMMA/C=CZRMMA.DOC,CZRMMA,SYSMAC/M
```

```
$$GET4  13-20     13-20#
$$SW08  35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6      35-6
        35-6      35-6      35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#
        35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#
        35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#
        35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#     35-6#
        35-6#     35-6#     35-6#     35-6#
$0FILL  35-4      35-4#     35-4*     35-4*
$40CAT  35-6      35-7
$APTHD   4-781     4-781#
$ASTAT  35-12     35-12
$ATY1   35-12#
$ATY3   35-5      35-12#
$ATY4   35-7      35-12#
$ATYC   35-12     35-12#
$AUTOB   5-0#      9-25*     9-62     11-27*    35-8      35-8      35-8
$BASE    5-0#      9-72     10-56     10-65*    11-47     12-2      12-34     12-63     12-91     12-196    12-238    12-263    12-314    12-332
        12-434    12-459    12-476    12-500    12-522    12-547    12-569    12-599    12-617    12-648    12-684    12-719    12-751    12-816
        12-870    12-901    12-945    12-974    12-:40    12-;07    12-;73    12-<39    12-<81    12-=13    12-=46    12->13    12->41    12->69
        12-?03    12-?37    12-?72    12-@06    12-@40    12-@89    12-A40    12-A88    12-B22    12-B59    12-B95    12-C72    12-D23    12-D83
        12-E28    12-E76    12-F22    12-F60    12-F99    12-G43    17-36     18-30     20-11     25-23     27-15     46-2
$BDADR   5-0#     46-8
$BDDAT   5-0#     12-127*   12-128*   12-130    12-135*   12-139*   12-144    12-149*   12-150*   12-154*   12-160*   12-164*   12-165*   12-167
        12-172*   12-176*   12-177*   12-179    12-184*   12-185*   12-188*   12-189*   12-191    12-250*   12-252*   12-256*   12-257*   12-276*
        12-282*   12-288*   12-295*   12-304*   12-309*   12-343*   12-348*   12-349*   12-354*   12-355*   12-360*   12-361*   12-366*   12-367*
        12-372*   12-373*   12-383*   12-388*   12-389*   12-394*   12-395*   12-400*   12-401*   12-406*   12-407*   12-412*   12-413*   12-418*
        12-419*   12-428*   12-430*   12-630*   12-631*   12-636*   12-637*   12-659*   12-660*   12-672*   12-673*   12-694*   12-695*   12-711*
        12-712*   12-729*   12-730*   12-742*   12-743*   12-884*   12-885*   12-889*   12-890*   12-922*   12-923*   12-929*   12-934*   12-957*
        12-958*   12-968*   12-969*   12-:17*   12-:18*   12-:19    12-:28*   12-:29*   12-:30    12-:84*   12-:85*   12-:86    12-:95*   12-:96*
        12-:97    12-;50*   12-;51*   12-;52    12-;61*   12-;62*   12-;63    12-<16*   12-<17*   12-<18    12-<27*   12-<28*   12-<29    12-=09*
        12-=38*   12-=68*   12-=69*   12-=70    12-B13*   12-B14*   12-B42*   12-B43*   12-C56*   12-F51*   12-F52*   12-F82*   12-F83*   21-49*
        21-50*    21-53     21-71*    21-77*    21-93*    21-110*   21-126*   21-154*   22-40*    22-41*    22-53*    22-54*    22-64*    22-65*
        22-67     22-85*    22-86*    22-87     22-107*   22-108*   22-109    22-140*   22-141*   22-142    22-156*   22-157*   22-158    22-170*
        22-171*   22-183*   22-184*   22-200*   22-201*   22-226*   22-227*   22-241*   22-242*   22-256*   22-257*   22-271*   22-272*   22-284*
        22-285*   22-314*   22-315*   22-329*   22-330*   22-344*   22-345*   22-361*   22-362*   22-387*   22-388*   22-402*   22-403*   22-417*
        22-418*   22-432*   22-433*   22-455*   22-456*   22-470*   22-471*   22-492*   22-493*   23-10*    23-34*    23-92*    23-93*    23-94
        23-120*   23-121*   23-122    23-137*   23-138    23-150*   23-151    24-39*    24-54*    26-37*    26-66*    26-67*    26-68     26-88*
        26-99*    26-100*   26-106    26-117    26-130*   26-143*   26-168*   26-182*   26-183*   26-196*   26-212*   28-31*    28-32*    28-34
        28-44*    28-53*    28-61     28-71*    28-72*    28-81*    28-82*    28-84     28-94*    28-103*   28-104*   28-106    28-116*   28-117*
        28-126*   28-127*   28-129    29-27*    29-41*    29-57*    29-71*    29-85*    29-99*    29-113*   30-35*    30-49*    30-65*    30-82*
        30-103*   30-120*   30-134*   30-159*   30-174*   30-189*   30-205*   30-222*   30-240*   30-254*   30-270*   31-16*    31-17*    31-19
        31-28*    31-29*    31-31     31-41*    31-50*    31-57*    31-60     31-69*    31-70*    31-72     31-81*    31-82*    31-84     31-94*
        31-103*   32-58*    32-76*    32-107*   32-108*   32-109    32-129*   32-130*   32-133    32-153*   32-170*   32-186*   32-209*   32-227*
        32-243*   32-257*   33-46*    33-62*    33-78*    33-98*    33-112*   46-1      46-6      46-8      46-10     46-11
$BELL    5-0#     35-7      35-7      35-7      35-8      35-8      35-8
$BIN    35-2      35-2#     35-2*     35-2*
$CDW1    5-0#
$CDW2    5-0#
$CHARC  35-5      35-5#     35-5*     35-5*     35-5*
```

D 7

23-136* 23-138 23-148* 23-149* 23-151 24-40* 24-55* 26-36* 26-47* 26-63* 26-68 26-87* 26-101* 26-102*

SEQ 0287

12-?03# 12-?03# 12-?37 12-?37# 12-?37# 12-?72 12-?72# 12-?72# 12-@06 12-@06# 12-@06# 12-@40 12-@40# 12-@40#

$TKQOU	35-8	35-8	35-8#	35-8*	35-8*	35-8*

```
$TKQSR  35-8    35-8    35-8    35-8#
$TKS     5-0#   35-5    35-5    35-8    35-8    35-8    35-8    35-8    35-8*   35-8*   35-8*   35-8*   35-8*   35-8*
$TKSRV  35-8    35-8#
$TMP0    5-0#   12-44*  12-47   12-=39* 12-=72* 17-41*  18-35*  25-29*  25-31   46-3    46-6    46-10
$TMP1    5-0#   10-13*  10-14   10-15   10-20   10-29*  10-30   10-31   10-33   10-44*  10-45   10-46   10-48   10-59*
        10-61   10-65   10-75*  10-77   10-81   10-96*  10-98   10-102* 10-103* 10-104* 10-105* 10-106* 10-107  10-118*
        10-119  10-125  10-127  10-128  10-130  10-135* 10-136  10-139  10-140  10-142  10-147  12-43*  12-51   12-84*
        17-42*  17-43   18-36*  18-37   24-24*  24-25*  24-37   24-39   25-28*  25-37   46-3    46-5    46-10
$TMP2    5-0#   12-85*  24-27*  24-28*  24-51   24-54   46-5
$TMP3    5-0#
$TMP4    5-0#
$TN      4-469#  4-479   12-2    12-2    12-2    12-2#   12-34   12-34   12-34   12-34#  12-63   12-63   12-63   12-63#
        12-91   12-91   12-91   12-91#  12-196  12-196  12-196  12-196# 12-238  12-238  12-238  12-238# 12-263  12-263
        12-263  12-263# 12-314  12-314  12-314  12-314# 12-332  12-332  12-332  12-332# 12-434  12-434  12-434  12-434#
        12-459  12-459  12-459  12-459# 12-476  12-476  12-476  12-476# 12-500  12-500  12-500  12-500# 12-522  12-522
        12-522  12-522# 12-547  12-547  12-547  12-547# 12-569  12-569  12-569  12-569# 12-599  12-599  12-599  12-599#
        12-617  12-617  12-617  12-617# 12-648  12-648  12-648  12-648# 12-684  12-684  12-684  12-684# 12-719  12-719
        12-719  12-719# 12-751  12-751  12-751  12-751# 12-816  12-816  12-816  12-816# 12-870  12-870  12-870  12-870#
        12-901  12-901  12-901  12-901# 12-945  12-945  12-945  12-945# 12-974  12-974  12-974  12-974# 12-:40  12-:40
        12-:40  12-:40# 12-;07  12-;07  12-;07  12-;07# 12-;73  12-;73  12-;73  12-;73# 12-<39  12-<39  12-<39  12-<39#
        12-<81  12-<81  12-<81  12-<81# 12-=13  12-=13  12-=13  12-=13# 12-=46  12-=46  12-=46  12-=46# 12->13  12->13
        12->13  12->13# 12->41  12->41  12->41  12->41# 12->69  12->69  12->69  12->69# 12-?03  12-?03  12-?03  12-?03#
        12-?37  12-?37  12-?37  12-?37# 12-?72  12-?72  12-?72  12-?72# 12-@06  12-@06  12-@06# 12-@40  12-@40
        12-@40  12-@40# 12-@89  12-@89  12-@89  12-@89# 12-A40  12-A40  12-A40  12-A40# 12-A88  12-A88  12-A88  12-A88#
        12-B22  12-B22  12-B22  12-B22# 12-B59  12-B59  12-B59  12-B59# 12-B95  12-B95  12-B95  12-B95# 12-C72  12-C72
        12-C72  12-C72# 12-D23  12-D23  12-D23  12-D23# 12-D83  12-D83  12-D83  12-D83# 12-E28  12-E28  12-E28  12-E28#
        12-E76  12-E76  12-E76  12-E76# 12-F22  12-F22  12-F22  12-F22# 12-F60  12-F60  12-F60  12-F60# 12-F99  12-F99
        12-F99  12-F99# 12-G43  12-G43  12-G43  12-G43# 35-6    35-6
$TPB     5-0#   35-5    35-5    35-5*
$TPFLG   5-0#   35-5    35-5    35-5
$TPS     5-0#   35-5    35-5    35-5
$TRAP    9-19   35-10#
$TRAP2  35-10   35-10#
$TRP    35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10
        35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10
        35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10
        35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10   35-10#  35-10#  35-10#  35-10#
        35-10#  35-10#  35-10#  35-10#  35-10#  35-10#  35-10#  35-10#  35-10#  35-10#
$TRPAD  35-10   35-10#
$TSTM    4-781#
$TSTNM   5-0#   11-37*  13-20*  35-6    35-6    35-6    35-6    35-6    35-6*   35-6*   35-7    35-7    35-7
$TTYIN  35-8    35-8    35-8    35-8    35-8    35-8#
$TYPBN  35-2#   35-10   35-10
$TYPDS  35-3#   35-10   35-10
$TYPE   35-5#   35-10   35-10   35-12
$TYPEC  35-5    35-5    35-5    35-5#   35-8
$TYPEX  35-5    35-5    35-5#
$TYPOC  35-4#   35-10   35-10
$TYPON  35-4    35-4#   35-10
$TYPOS  35-4#   35-10
$UNIT    5-0#   11-42*  14-23   21-51
$UNITM   4-781#
$USWR    5-0#
$VECT1   5-0#   10-69   10-81*  10-85   10-107* 12-755  12-761  12-820  12-826
$VECT2   5-0#
$XOFF   35-5    35-5
```

SEQ 0293

J 7

$XON 35-5 35-5

```
$XTSTR  35-6#
.$ASTA  35-12     35-12
.$X      4-781     4-781#
A16      4-720#   12-=98
A17      4-719#   12-=98
ABASE    4-771#    5-0        5-0
ACDW1    5-0       5-0
ACDW2    5-0       5-0
ACKSTS  12-452    15-151    29-15#
ACPUOP   5-0       5-0
ADDW0    5-0       5-0
ADDW1    5-0       5-0
ADDW10   5-0
ADDW11   5-0
ADDW12   5-0
ADDW13   5-0
ADDW14   5-0
ADDW15   5-0
ADDW2    5-0       5-0
ADDW3    5-0       5-0
ADDW4    5-0       5-0
ADDW5    5-0       5-0
ADDW6    5-0       5-0
ADDW7    5-0       5-0
ADDW8    5-0
ADDW9    5-0
ADEVCT   5-0       5-0
ADEVM    5-0       5-0
ADR      12-2      12-2#     12-2#     12-34     12-34#    12-34#    12-63     12-63#    12-63#    12-91     12-91#    12-91#    12-196    12-196#
         12-196#   12-238    12-238#   12-238#   12-263    12-263#   12-263#   12-314    12-314#   12-314#   12-332    12-332#   12-332#   12-434
         12-434#   12-434#   12-459    12-459#   12-459#   12-476    12-476#   12-476#   12-500    12-500#   12-500#   12-522    12-522#   12-522#
         12-547    12-547#   12-547#   12-569    12-569#   12-569#   12-599    12-599#   12-599#   12-617    12-617#   12-617#   12-648    12-648#
         12-648#   12-684    12-684#   12-684#   12-719    12-719#   12-719#   12-751    12-751#   12-751#   12-816    12-816#   12-816#   12-870
         12-870#   12-870#   12-901    12-901#   12-901#   12-945    12-945#   12-945#   12-974    12-974#   12-974#   12-:40    12-:40#   12-:40#
         12-;07    12-;07#   12-;07#   12-;73    12-;73#   12-;73#   12-<39    12-<39#   12-<39#   12-<81    12-<81#   12-<81#   12-=13    12-=13#
         12-=13#   12-=46    12-=46#   12-=46#   12->13    12->13#   12->13#   12->41    12->41#   12->41#   12->69    12->69#   12->69#   12-?03
         12-?03#   12-?03#   12-?37    12-?37#   12-?37#   12-?72    12-?72#   12-?72#   12-@06    12-@06#   12-@06#   12-@40    12-@40#   12-@40#
         12-@89    12-@89#   12-@89#   12-A40    12-A40#   12-A40#   12-A88    12-A88#   12-A88#   12-B22    12-B22#   12-B22#   12-B59    12-B59#
         12-B59#   12-B95    12-B95#   12-B95#   12-C72    12-C72#   12-C72#   12-D23    12-D23#   12-D23#   12-D83    12-D83#   12-D83#   12-E28
         12-E28#   12-E28#   12-E76    12-E76#   12-E76#   12-F22    12-F22#   12-F22#   12-F60    12-F60#   12-F60#   12-F99    12-F99#   12-F99#
         12-G43    12-G43#   12-G43#
AENV     5-0       5-0
AENVM    5-0       5-0
AFATAL   5-0       5-0
ALL     10-121    36-4#
AMADR1   5-0       5-0
AMADR2   5-0       5-0
AMADR3   5-0       5-0
AMADR4   5-0       5-0
AMAMS1   5-0       5-0
AMAMS2   5-0       5-0
AMAMS3   5-0       5-0
AMAMS4   5-0       5-0
AMSGAD   5-0       5-0
AMSGLG   5-0       5-0
AMSGTY   5-0       5-0
```

AMTYP1 5-0 5-0

L 7

SEQ 0295

12-231# 12-235 12-235# 12-235# 12-248 12-248# 12-269 N 7 12-269# 12-323 12-323# 12-326 12-326# 12-326# 12-329

SEQ 0297

```
EFT111   7-225    7-228   43-5#
EFT114   7-234   43-7#
EFT2     7-427    7-430    7-514    7-517   43-2#
EFT223   7-448    7-529   43-8#
EFT336   7-352    7-682    7-688    7-691   43-10#
EFT337   7-685   43-11#
EFT344   7-700   43-12#
EFT353   7-721   43-14#
EH1     41-1     45-1#
EH110   41-4     45-2#
EH111   41-5     45-3#
EH114   41-7     45-5#
EH223   41-8     45-6#
EH256   41-9     45-7#
EH336   41-11    45-10#
EH337   41-12    45-11#
EH344   41-13    45-15#
EH353   41-15    45-18#
EHT1     7-4      7-7      7-10     7-13     7-16     7-19     7-25     7-28     7-31     7-34     7-37     7-40     7-43     7-46
         7-49     7-52     7-55     7-58     7-61     7-64     7-67     7-70     7-73     7-76     7-79     7-82     7-85     7-88
         7-91     7-94     7-97     7-100    7-103    7-106    7-109    7-112    7-115    7-118    7-121    7-124    7-127    7-131
         7-134    7-137    7-140    7-143    7-147    7-151    7-155    7-161    7-164    7-167    7-170    7-173    7-176    7-179
         7-183    7-186    7-189    7-192    7-195    7-198    7-201    7-204    7-207    7-210    7-213    7-216    7-219    7-237
         7-240    7-243    7-246    7-249    7-252    7-255    7-258    7-261    7-264    7-267    7-270    7-273    7-276    7-279
         7-282    7-285    7-288    7-291    7-294    7-297    7-300    7-303    7-306    7-309    7-312    7-315    7-318    7-321
         7-324    7-327    7-330    7-333    7-336    7-339    7-342    7-345    7-348    7-355    7-358    7-361    7-364    7-370
         7-373    7-376    7-379    7-382    7-385    7-388    7-391    7-394    7-397    7-400    7-403    7-406    7-409    7-412
         7-415    7-418    7-421    7-424    7-433    7-436    7-439    7-442    7-445    7-505    7-508    7-511    7-520    7-523
         7-526    7-532    7-535    7-538    7-541    7-544    7-547    7-550    7-553    7-557    7-560    7-564    7-567    7-570
         7-574    7-578    7-582    7-587    7-591    7-595    7-598    7-601    7-604    7-607    7-610    7-613    7-616    7-619
         7-622    7-625    7-628    7-631    7-634    7-637    7-640    7-643    7-646    7-649    7-652    7-655    7-658    7-661
         7-664    7-667    7-670    7-673    7-676    7-679    7-694    7-697    7-703    7-706    7-709    7-712    7-715    7-724
        41-1#
EHT110   7-222   41-4#
EHT111   7-225    7-228   41-5#
EHT114   7-234   41-7#
EHT2     7-427    7-430    7-514    7-517   41-2#
EHT223   7-448   41-8#
EHT256   7-529   41-9#
EHT336   7-352    7-682    7-688    7-691   41-11#
EHT337   7-685   41-12#
EHT344   7-700   41-13#
EHT353   7-721   41-15#
EMS1    40-3     44-3#
EMS10   40-7     40-11    44-10#
EMS100  40-39    40-71    40-176   40-196   44-66#
EMS101  40-40    40-59    40-174   40-197   44-67#
EMS102  40-177   40-178   40-181   40-182   40-183   40-184   40-186   40-187   40-188   40-189   40-234   40-235   44-68#
EMS103  40-56    40-57    40-135   40-141   40-196   40-197   40-198   40-203   40-204   40-205   40-206   40-207   40-208   40-209
        40-210   40-211   40-213   40-214   40-215   40-219   40-220   44-69#
EMS104  40-136   40-180   40-203   44-70#
EMS105  40-204   44-71#
EMS106  40-212   40-232   44-72#
EMS11   40-12    40-15    40-16    40-17    40-18    40-19    40-20    40-21    40-22    40-23    40-24    40-25    40-26    40-27
        40-28    40-29    40-30    40-31    40-32    40-33    40-43    40-177   44-11#
EMS110  40-9     40-74    40-77    44-73#
```

```
EMS112  40-74   44-75#
EMS113  40-75   40-76   40-78   40-127  44-77#
EMS114  40-77   44-78#
EMS115  40-44   40-45   40-48   40-50   40-51   40-52   40-54   40-55   40-60   40-61   40-63   40-66   40-67   40-70
        40-73   40-77   40-107  40-108  40-109  40-110  40-134  40-137  40-139  40-140  40-142  40-179  40-182  40-183
        40-185  40-186  40-187  40-188  40-189  40-190  40-192  40-193  40-194  40-195  40-201  40-205  40-221  40-222
        40-223  40-232  40-234  44-79#
EMS116  40-74   40-77   40-191  40-225  44-80#
EMS117  40-77   40-143  40-148  40-149  40-178  40-185  40-201  40-225  40-228  40-229  40-230  40-231  44-81#
EMS12   40-12   44-12#
EMS120  40-78   44-82#
EMS121  40-79   40-88   40-97   40-99   44-83#
EMS122  40-80   40-89   44-84#
EMS123  40-81   40-88   40-89   40-90   40-90   40-91   40-92   40-93   40-94   40-95   40-96   40-98   40-126  44-85#
EMS124  40-56   40-82   40-91   40-101  44-86#
EMS125  40-83   40-92   40-102  40-169  44-87#
EMS126  40-84   40-93   40-103  44-88#
EMS127  44-89#
EMS13   40-13   40-42   40-216  44-13#
EMS130  40-86   40-95   40-106  44-90#
EMS131  40-87   40-96   40-104  44-91#
EMS132  40-79   40-80   40-81   40-82   40-83   40-84   40-85   40-86   40-87   40-88   40-89   40-90   40-91   40-92
        40-93   40-94   40-95   40-96   40-97   40-98   40-99   40-100  40-101  40-102  40-103  40-104  40-105  40-106
        40-125  40-126  40-168  40-169  44-92#
EMS133  40-79   40-80   40-81   40-82   40-83   40-84   40-85   40-86   40-87   40-125  44-93#
EMS134  40-88   40-89   40-90   40-91   40-92   40-93   40-94   40-95   40-96   40-126  44-94#
EMS135  40-97   40-98   44-95#
EMS136  40-99   40-100  40-101  40-102  40-103  40-104  40-105  40-106  44-96#
EMS137  40-85   40-94   40-100  44-97#
EMS14   40-13   44-14#
EMS140  40-44   40-52   40-60   40-107  40-117  40-137  40-186  40-192  44-98#
EMS141  40-35   40-44   40-45   40-60   40-61   40-66   40-107  40-129  40-137  40-187  40-188  40-193  40-222  44-99#
EMS142  40-47   40-65   40-108  40-140  40-184  40-202  44-100#
EMS143  40-47   40-65   40-73   40-108  40-109  40-110  40-133  40-140  40-184  40-202  40-205  44-101#
EMS144  40-36   40-73   40-109  40-110  40-205  44-102#
EMS145  40-36   40-73   40-109  40-110  40-205  44-103#
EMS146  40-109  40-185  40-190  40-201  44-104#
EMS147  40-69   40-112  40-113  40-173  40-217  44-105#
EMS15   40-13   40-42   40-216  44-15#
EMS150  40-50   40-51   40-54   40-63   40-67   40-111   )-134  40-138  40-182  40-183  40-194  40-195  40-223  40-234
        44-106#
EMS151  40-69   40-112  40-113  40-173  40-217  44-107#
EMS152  40-50   40-51   40-54   40-63   40-111  40-134  40-182  40-183  40-194  40-195  40-223  44-108#
EMS153  40-111  44-109#
EMS154  40-111  40-112  40-122  40-123  40-124  44-110#
EMS155  40-112  40-124  44-111#
EMS156  40-113  40-114  40-115  40-116  40-173  44-112#
EMS157  40-113  40-123  44-113#
EMS16   40-13   44-16#
EMS160  40-114  40-115  40-116  44-114#
EMS161  40-114  44-115#
EMS162  44-116#
EMS163  40-58   40-59   40-60   40-61   40-62   40-63   40-64   40-65   40-66   40-67   40-68   40-69   40-70   40-71
        40-72   40-73   40-117  40-143  44-117#
EMS164  40-45   44-118#
EMS165  40-56   40-57   40-71   40-72   40-172  40-196  40-197  40-198  44-120#
```

EMS167	40-8	40-49	40-68	40-122	40-123	40-124	40-168	40-169	40-170	40-175	40-235	44-122#	
EMS17	40-14	44-17#											
EMS170	40-9	44-123#											
EMS171	40-57	40-105	40-125	40-126	44-124#								
EMS172	40-215	44-125#											
EMS173	40-214	44-126#											
EMS174	40-216	44-127#											
EMS175	40-216	40-217	40-218	40-219	44-128#								
EMS176	44-129#												
EMS177	40-127	44-130#											
EMS2	40-4	40-5	44-4#										
EMS20	40-34	40-58	40-110	40-118	40-119	40-120	40-121	40-212	40-232	40-238	44-18#		
EMS200	40-128	44-131#											
EMS201	40-128	44-132#											
EMS202	40-129	40-130	40-131	40-132	40-133	40-134	40-136	40-180	44-133#				
EMS203	40-129	40-130	40-131	40-132	40-133	40-174	40-175	40-176	40-237	44-134#			
EMS204	40-131	44-135#											
EMS205	40-55	40-142	40-179	40-233	44-136#								
EMS206	40-55	40-142	40-170	40-179	40-233	44-137#							
EMS207	40-168	40-170	44-138#										
EMS21	40-15	44-19#											
EMS210	40-169	40-170	40-173	44-139#									
EMS211	40-170	44-140#											
EMS212	40-171	40-172	44-141#										
EMS213	40-171	44-142#											
EMS214	40-55	44-143#											
EMS215	40-236	44-144#											
EMS216	40-237	44-145#											
EMS217	40-237	44-146#											
EMS22	40-16	44-20#											
EMS220	40-179	44-147#											
EMS221	40-231	40-238	44-148#										
EMS222	40-233	44-149#											
EMS223	40-232	44-150#											
EMS224	40-121	44-151#											
EMS23	40-17	44-21#											
EMS24	40-18	44-22#											
EMS25	40-19	40-115	44-23#										
EMS26	40-20	40-116	40-173	44-24#									
EMS27	40-16	40-19	40-20	40-115	40-116	40-173	44-25#						
EMS3	40-4	40-76	44-5#										
EMS30	40-17	40-18	40-21	40-22	40-23	40-24	40-25	40-26	40-27	40-28	40-29	40-30	40-31	40-32
	40-33	40-43	40-129	40-130	40-131	40-132	40-133	40-144	40-145	40-146	40-147	40-174	40-175	40-176
	40-177	40-199	44-26#											
EMS31	40-21	40-146	40-207	40-208	44-27#									
EMS32	40-22	44-28#												
EMS33	40-23	40-219	44-29#											
EMS34	40-24	40-225	40-226	40-227	44-30#									
EMS35	40-25	40-216	44-31#											
EMS36	40-26	40-213	40-214	40-215	44-32#									
EMS37	40-27	44-33#												
EMS4	40-5	40-75	44-6#											
EMS40	40-28	40-220	44-34#											
EMS41	40-29	40-190	40-191	44-35#										
EMS42	40-30	40-206	44-36#											
EMS43	40-31	40-209	44-37#											

EMS44 40-32 40-211 40-230 44-38#

SEQ 0311

EMT116 7-240 40-80#
EMT117 7-243 40-81#
EMT12 7-31 40-12#
EMT120 7-246 40-82#
EMT121 7-249 40-83#
EMT122 7-252 40-84#
EMT123 7-255 40-85#
EMT124 7-258 40-86#
EMT125 7-261 40-87#
EMT126 7-264 40-88#
EMT127 7-267 40-89#
EMT13 7-34 40-13#
EMT130 7-270 40-90#
EMT131 7-273 40-91#
EMT132 7-276 40-92#
EMT133 7-279 40-93#
EMT134 7-282 40-94#
EMT135 7-285 40-95#
EMT136 7-288 40-96#
EMT137 7-291 40-97#
EMT14 7-37 40-14#
EMT140 7-294 40-98#
EMT141 7-297 40-99#
EMT142 7-300 40-100#
EMT143 7-303 40-101#
EMT144 7-306 40-102#
EMT145 7-309 40-103#
EMT146 7-312 40-104#
EMT147 7-315 40-105#
EMT15 7-40 40-15#
EMT150 7-318 40-106#
EMT151 7-321 40-107#
EMT152 7-324 40-108#
EMT153 7-327 40-109#
EMT154 7-330 40-110#
EMT155 7-333 40-111#
EMT156 7-336 40-112#
EMT157 7-339 40-113#
EMT16 7-43 40-16#
EMT160 7-342 40-114#
EMT161 7-345 40-115#
EMT162 7-348 40-116#
EMT163 40-117#
EMT164 7-355 40-118#
EMT165 7-358 40-119#
EMT166 7-361 40-120#
EMT167 7-364 40-121#
EMT17 7-46 40-17#
EMT170 40-122#
EMT171 7-370 40-123#
EMT172 7-373 40-124#
EMT173 7-376 40-125#
EMT174 7-379 40-126#
EMT175 7-382 40-127#
EMT176 7-385 40-128#
EMT177 7-388 40-129#

```
EMT20    7-49    40-18#
EMT200   7-391   40-130#
EMT201   7-394   40-131#
EMT202   7-397   40-132#
EMT203   7-400   40-133#
EMT204   7-403   40-134#
EMT205   7-406   40-135#
EMT206   7-409   40-136#
EMT207   7-412   40-137#
EMT21    7-52    40-19#
EMT210   7-415   40-138#
EMT211   7-418   40-139#
EMT212   7-421   40-140#
EMT213   7-424   40-141#
EMT214   7-427   40-142#
EMT215   7-430   40-143#
EMT216   7-433   40-144#
EMT217   7-436   40-145#
EMT22    7-55    40-20#
EMT220   7-439   40-146#
EMT221   7-442   40-147#
EMT222   7-445   40-148#
EMT223   7-448   40-149#
EMT224  40-150#
EMT225  40-151#
EMT226  40-152#
EMT227  40-153#
EMT23    7-58    40-21#
EMT230  40-154#
EMT231  40-155#
EMT232  40-156#
EMT233  40-157#
EMT234  40-158#
EMT235  40-159#
EMT236  40-160#
EMT237  40-161#
EMT24    7-61    40-22#
EMT240  40-162#
EMT241  40-163#
EMT242  40-164#
EMT243  40-165#
EMT244  40-166#
EMT245  40-167#
EMT246   7-505   40-168#
EMT247   7-508   40-169#
EMT25    7-64    40-23#
EMT250   7-511   40-170#
EMT251   7-514   40-171#
EMT252   7-517   40-172#
EMT253   7-520   40-173#
EMT254   7-523   40-174#
EMT255   7-526   40-175#
EMT256   7-529   40-176#
EMT257   7-532   40-177#
EMT26    7-67    40-24#
EMT260   7-535   40-178#
```

EMT262 7-541 40-180#
EMT263 7-544 40-181#
EMT264 7-547 40-182#
EMT265 7-550 40-183#
EMT266 7-553 40-184#
EMT267 7-557 40-185#
EMT27 7-70 40-25#
EMT270 7-560 40-186#
EMT271 7-564 40-187#
EMT272 7-567 40-188#
EMT273 7-570 40-189#
EMT274 7-574 40-190#
EMT275 7-578 40-191#
EMT276 7-582 40-192#
EMT277 7-587 40-193#
EMT3 7-10 40-5#
EMT30 7-73 40-26#
EMT300 7-591 40-194#
EMT301 7-595 40-195#
EMT302 7-598 40-196#
EMT303 7-601 40-197#
EMT304 7-604 40-198#
EMT305 7-607 40-199#
EMT306 7-610 40-200#
EMT307 7-613 40-201#
EMT31 7-76 40-27#
EMT310 7-616 40-202#
EMT311 7-619 40-203#
EMT312 7-622 40-204#
EMT313 7-625 40-205#
EMT314 7-628 40-206#
EMT315 7-631 40-207#
EMT316 7-634 40-208#
EMT317 7-637 40-209#
EMT32 7-79 40-28#
EMT320 7-640 40-210#
EMT321 7-643 40-211#
EMT322 7-646 40-212#
EMT323 7-649 40-213#
EMT324 7-652 40-214#
EMT325 7-655 40-215#
EMT326 7-658 40-216#
EMT327 7-661 40-217#
EMT33 7-82 40-29#
EMT330 7-664 40-218#
EMT331 7-667 40-219#
EMT332 7-670 40-220#
EMT333 7-673 40-221#
EMT334 7-676 40-222#
EMT335 7-679 40-223#
EMT336 7-352 7-682 40-224#
EMT337 7-685 40-225#
EMT34 7-85 40-30#
EMT340 7-688 40-226#
EMT341 7-691 40-227#
EMT342 7-694 40-228#

12-38 12-39 **12-40*** **12-41*** 12-45* 12-46* 12-57* L 9 12-58* 12-=52 12-=53 12-=54* 12-=55* 12->10* 12->11*

SEQ 0321

N 9
a09# 12-a09# 12-a20 12-a20# 12-a20# 12-a24 12-a24# 12-a24#
SEQ 0323

12-?96# 12-?96# 12-?98 12-?98# 12-?98# 12-a09

```
HELP    10-37   48-7#
HT       4-483#  14-62   35-5    35-5
IAE      4-570#  12-;49  12-;51  12-@74  12-A23  12-A72  12-E13  12-E59  12-F06  22-283  23-89   23-105  23-132  26-47
        26-67   30-24   30-78   30-81   32-88   32-108  37-59   37-69   37-77   37-78   37-81   37-82   37-85   37-86
IDXMSK   4-709#  17-51   17-54   18-45   18-48
IE       4-722#   4-755#  12-779  12-840
ILF      4-580#  12-:16  12-:18  22-155  22-157  29-51   29-111  29-115  30-24   30-45   30-48   37-58   37-67   37-68
        37-70   37-71   37-72   37-73   37-74   37-75   37-76   37-79   37-80   37-83   37-84   37-87   37-88
ILF02    4-500#
ILF24    4-510#
ILF26    4-511#
ILF30    4-515#
ILF32    4-515#
ILF34    4-515#
ILF36    4-515#
ILF40    4-515#
ILF42    4-515#
ILF44    4-515#
ILF46    4-515#
ILF54    4-518#
ILF56    4-519#
ILF64    4-522#
ILF66    4-523#
ILF74    4-526#
ILF76    4-527#  12-:35  12-;02  12-;68  12-<34  12-<77
ILR      4-579#  12-=69  12-=97  29-51   29-97   29-101  30-232  30-236  30-239
ILRG50   4-707#
ILRG52   4-707#
ILRG54   4-707#
ILRG56   4-707#
ILRG60   4-707#
ILRG62   4-707#
ILRG64   4-707#
ILRG66   4-707#
ILRG70   4-707#
ILRG72   4-707#
ILRG74   4-707#
ILRG76   4-707#
IOTVEC   4-483#  9-19*   9-19*
IPCK0    4-759#
IPCK1    4-758#
IPCK2    4-757#
IPCK3    4-756#
IR       4-735#  12-142  28-58
IVC      4-676#  12-515  12-:83  12-:85  12-B10  12-B12  12-B14  12-B17  12-F48  12-F50  12-F52  12-F55  22-198  22-199
        26-85   26-208  30-92   30-99   30-102  30-171  32-125  32-149  32-152  32-239  33-58   37-58   37-59   37-60
        37-62   37-63   37-64   37-67   37-68   37-69   37-70   37-71   37-72   37-73   37-74   37-75   37-76   37-77
        37-78   37-79   37-80   37-81   37-82   37-83   37-84   37-85   37-86   37-87   37-88
LBC      4-678#  12-426  12-429  12-430
LBT      4-556#  12-274  22-358  23-91   23-93
LF       4-483#  14-60   35-5    35-5    36-14   36-16   44-79
LODEV    9-76    36-24#
LS       4-633#  12-165  28-82   31-70
LSC      4-677#  12-577
LST      4-634#  12-165  28-82   31-70
LSTRK    6-0#    11-46*  11-54*  12-@95  12-A34  12-E34  12-E70  23-62   26-53   32-94
```

MCLK 4-607#

SEQ 0327

RMAS 4-693# 12-702 12-774

I 10

SEQ 0331

22-358 23-92 26-91 26-146 26-155 26-162 26-166 26-168 26-177 26-181 26-183 26-192 26-194 26-196

RMWC1 6-0# 23-10 23-23 23-44 23-115 46-14

M 10

SEQ 0335

STSH4	41-1	41-2	41-8	41-9	41-11	41-12	41-13	45-26#						
SW0	4-483#													
SW00	4-483	4-483#												
SW01	4-483	4-483#												
SW02	4-483	4-483#												
SW03	4-483	4-483#												
SW04	4-483	4-483#												
SW05	4-483	4-483#												
SW06	4-483	4-483#												
SW07	4-483	4-483#												
SW08	4-483	4-483#												
SW09	4-483	4-483#												
SW1	4-483#													
SW10	4-483#													
SW11	4-483#													
SW12	4-483#													
SW13	4-483#	14-17												
SW14	4-483#													
SW15	4-483#													
SW2	4-483#													
SW3	4-483#													
SW4	4-483#													
SW5	4-483#													
SW6	4-483#													
SW7	4-483#													
SW8	4-483#													
SW9	4-483#													
SWR	5-0#	9-19	9-19	9-19*	9-19*	9-19*	9-25	11-27	14-17	35-6	35-6	35-6	35-6	35-6
	35-7	35-7	35-7	35-7	35-8	35-8	35-8*	35-11	35-11*					
SWREG	4-776#	9-19	9-25	11-27	35-8	35-8	35-8							
TA1	4-536#													
TA16	4-532#	11-54												
TA2	4-535#	11-54												
TA4	4-534#	11-46												
TA8	4-533#													
TADMSK	4-546#	23-66												
TAG	4-663#													
TAGADR	4-782#	5-0												
TAP	4-640#													
TBITVE	4-483#													
TIMOUT	12-321	12-444	12-465	12-489	12-510	12-536	12-558	12-587	12-623	12-781	12-842	12-876	12-917	12-:10
	12-:77	12-;43	12-<09	12-<69	12->28	12->56	12->85	12-?20	12-?54	12-?88	12-@22	12-@65	12-A14	12-A64
	12-B46	12-B74	12-B82	12-C13	12-C89	12-C99	12-D43	12-D55	12-E03	12-E49	12-E96	12-F42	12-F86	12-G16
	12-G23	12-G58	15-136	15-191	20-8#									
TKVEC	4-483#	35-8*	35-8*											
TPVEC	4-483#													
TRAPVE	4-483#	9-19*	9-19*											
TRE	4-716#	12-242	12-252	21-92	21-109	21-125	22-106	22-108	23-12					
TRTVEC	4-483#													
TST	4-664#	12-178	28-105	31-83										
TST1	12-2#	35-6												
TST10	12-314#	35-6												
TST11	12-332#	35-6												
TST12	12-434#	35-6												
TST13	12-459#	35-6												
TST14	12-476#	35-6												

TST15 12-500# 35-6

D 11

SEQ 0339

12-434 12-459 12-476 12-500 12-522 12-547 12-569 12-599 12-617 12-648 12-684 12-719 12-751 12-816

12-:40 12-;07 12-;73 12-<39 12-<81 12-=13 12-=46 J 11 12->13 12->41 12->69 12-?03 12-?37 12-?72 12-@06
SEQ 0345

7-646 7-649 7-652 7-655 7-658 7-661 7-664 L 11 7-667 7-670 7-673 7-676 7-679 7-682 7-685

SEQ 0347